سلسلة تربويات الحاسوب
استخدام الحاسوب وتكنولوجيا
المعلومات في التربية
- ٢ -

إعداد وإنتاج برمجيات



تأليف
دكتور/ إبراهيم عبد الوكيل الفار
أستاذ المناهج وطرق تعليم الرياضيات والحاسوب المشارك
كلية التربية . جامعة طنطا

الناشر



الدلتا لتكنولوجيا الحاسبات
٤٠ شارع مسجد الرضوان - طنطا

الطبعة الثانية

[illegible]

إعداد وإنتاج برمجيات



إبراهيم عبد الوكيل الفار
إعداد وإنتاج برمجيات الوسائط المتعددة التفاعلية- تأليف د. إبراهيم عبد الوكيل الفار
- طنطا - الدلتا لتكنولوجيا الحاسبات - ١٩٩٨
٢٨٥ صفحة : أبيض ٢٤ سم - سلسلة تربويات الحاسوب : استخدام الحاسوب وتكنولوجيا المعلومات في التربية (٢)
ببلبوجرافية : ص ٢٧٣-٢٨٤
رقم الإيداع بدار الكتب والوثائق المصرية ١٥٤٥٨ / ٩٨
الترقيم الدولي (تدمك) ISBN ‏٩-٨٥٦٣-١٩-٩٧٧
١- برمجة الحاسبات - الوسائط المتعددة - التعليم المبرمج - تصميم التعليم
٢- العنوان ٣- السلسلة

الصف والإخراج الفني للمؤلف



وعلم الإنسان ما لم يعلم



مقدمة

تمر عملية إنتاج البرمجيات التعليمية *Courseware* المختلفة بمراحل عديدة قبل أن تظهر بالصورة التي نراها. وتتميز البرمجيات التعليمية المقدمة بواسطة الحاسوب في أنها تتمتع باستقلال نسبى عن المعلم بعكس ما يحدث عند استخدام الكتب المدرسية ، فتخضع الكتب عادة لتحكم المعلم الذي يقوم بالاختيار والتعديل والحذف وإعادة الترتيب . كما يمارس التلميذ مزيدا من إعادة الترتيب والحذف الاختياري والتركيز على بعض الموضوعات دون غيرها عند المراجعة ، ومثل هذه المعالجات الخارجية قد تكون صعبة أو مستحيله بالنسبة للبرمجيات التعليمية . ومن هنا نرى أن استخدام مثل هذه الدروس المبرمجة يصاحبه تغيير أساسي في قواعد التحكم التعليمي *Instructional Control* وهذا يتطلب بالضرورة وضع قواعد جديدة في تصميم هذه البرمجيات وإعدادها وتقويمها.

سوف نحاول أن نقدم هنا ومن خلال هذا الباب بعض الملامح الأساسية التي ينبغي أن تتمتع بها البرمجيات التعليمية عند إعدادها ؛ حتى يستطيع أن يسترشد بها المعلم أو أي شخص آخر يفكر في إنتاج برمجية تعليمية عالية الجودة ، وكذا نقدم استمارة لتقويم البرمجيات التعليمية لمساعدة المعلمين وأولياء الأمور وكل من يهمه الأمر على اختيار البرمجيات التعليمية المستهدفة ، ولتقويم البرمجيات التعليمية التي يرغبون في إنتاجها على أسس علمية صحيحة . ونقدم في النهاية بشيء من التفصيل دورة إنتاج البرمجية التعليمية.

إن بعض البرمجيات التعليمية المتوافرة حاليا والمباعة بشكل تجارى تعوزها خصائص علمية كثيرة في تصميمها ، وقد يرجع ذلك إلى أن بعض القائمين على إنتاجها قد لا تتوافر لديهم الخبرة التربوية الكافية ، ولم يسبق لهم إعداد درس أو حتى القيام بالتدريس على الإطلاق ، في حين ، قد تتوافر لدى بعضهم الآخر مثل هذه الخبرات التربوية ، ولكن لا تتوافر لديه الخبرات الكافية المتعلقة باستخدام الحاسوب ومعرفة إمكاناته. وقد يشكك بعض المهتمون بالعملية التعليمية في جدوى استخدام البرمجيات التعليمية ؛ لضعف

البرمجيات التي شاهدوها أو لشعورهم بأن العائد الفعلي من وراء اســتخدامها قد لا يضاهى ما ينفق على شرائها ، أو أنه يمكن تحقيق نفس العائد بوســائل أخرى وبتكاليف أقل.

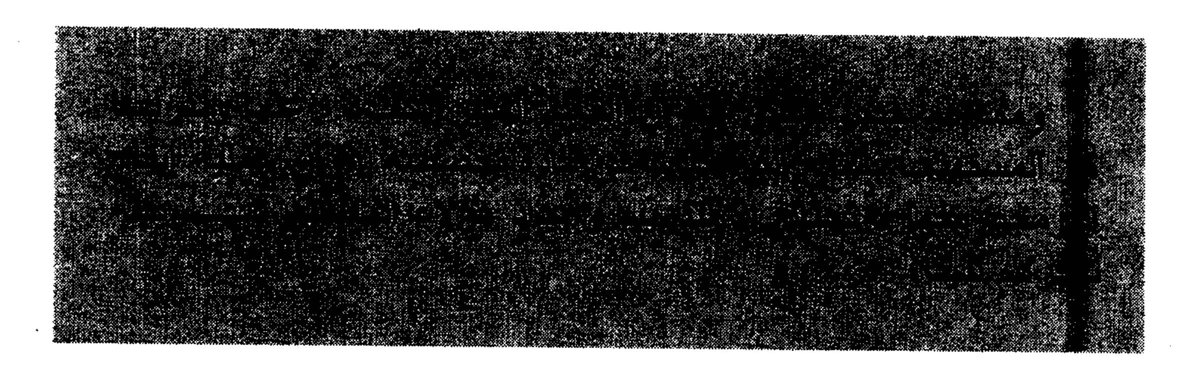
وكما أن هناك بعض البرمجيات التي يعوزها الأساس العلمي فإن هنــاك بعض البرامج المتوافرة التي تتمتع بخصائص علمية عديدة جيدة – من حيث التصميم والعرض والإخراج ، ولا يعقل أن نحجم عن استخدام البرمجيــات التعليمية لأن هناك بعضها يمكن وصفها بأنها غير جيدة ، ولكن مــا ينبغي عمله – هو **أن نفرق بين العمل الجيد والعمل غير الجيد ، وأن نعــرف مــا ينبغي أن تكون علية البرمجية التعليمية الجيدة ، ومن ثم يمكننا أن نوجــه من يعنيه الأمر لمراعاة بعض الاعتبارات الأساسية التي ينبغي الأخــذ بــها عند تصميم ووضع البرمجيات .**

تمر عملية إعداد البرمجيات التعليمية بعدة مراحل ، قبل أن تخــرج بالشكل النهائي الذي تعرض به ، وقد يقوم بهذه العملية مجموعة مختلفة مــن الأفراد أو المعلمين الذين تتوافر لديهم خبرات محددة ، وفيما يلي عرض لهذه المراحل :

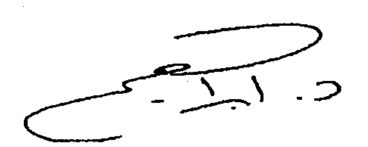
- **مرحلة التخطيط والتصميم *planing & Design***
- **مرحلة كتابة السيناريو *Scenario***
- **مرحلة التنفيذ *Executing***
- **مرحلة التجريب والتطوير *Development***

علي ألا نبخل بالوقت الذي يستغرق في مراحــل التصميــم والإعــداد وكتابة السيناريو ، فكلما كان السيناريو معدا بطريقة واضحة ومفصلة آخــذا في الاعتبار جميع الاحتمالات الممكنة ؛ كان الوقت المستغرق فــي عمليــة تنفيذ البرمجية أقل بكثير.

ومن الاختيارات المفضلة والمنشودة أن يكون معد سيناريو البرمجية هــو نفس المعلم الذي سيقوم بتنفيذ البرمجية على الحاسوب ، وهو نفسه الذي قــام بتصميم البرمجية ، وتجميع ما يلزم إنتاجها من مواد علمية وتعليمية وأنشــطة ومفردات اختبار ، ... الخ.



ينبغي أن تعرض البرمجية التي يتم إعدادها علي عدد من الموجهين والمعلمين ، وكذا خبراء المناهج وطرق التدريس ، وأساتذة علم النفس التربوي ، إضافة إلى عرضها عمليا علي عينة من التلاميذ مأخوذة من المجتمع الأصلي الذي ستطبق فيه هذه البرمجية وعلى ضوء مقترحات الموجهين والمعلمين وخبراء المناهج وطرق التدريس ، وكذا أساتذة علم النفس التربوي ، وفي ضوء مواقف عينة التلاميذ والطلاب ؛ يجري مزيدا من التعديلات على البرمجية إذا لزم لأمر ، إلى أن تصل إلى مستوى يسمح بنشرها وتعميمها على نطاق واسع ، وكثيرا ما تخضع البرمجيات التعليمية المعدة لنوع من التقويم بواسطة أدوات خاصة لتقويم البرمجيات ، والتي توضح نواحي القوة ونواحي الضعف بها ، حيث يمكن الاسترشاد بها عند محاولة اقتناء بعض البرامج من ناحية والتي تعتبر مفيدة بالنسبة لمنتجي البرمجيات من ناحية أخري ، عند إجراء أي تطوير لها في المستقبل. **ولذا كان من الضروري العمل علي تطوير أداة يمكن الاستفادة منها في هذه الأغراض ، وسوف نعرض خلال الفصل الثامن قائمة بنود لتقويم تلك البرمجيات ، والتي تعتبر أداة هامة في أيدي المعلمين للحكم علي مدي مناسبة البرمجيات التعليمية سواء كانت منتجة بواسطتهم أو مشتراة.**



د. إبراهيم عبد الوكيل الفار
أكتوبر ١٩٩٨

فهرس المحتويات

(تابع) الفصل الأول
مرحلة تصميم وإعداد البرمجية

الفصل الثاني
مرحلة كتابة سيناريو البرمجية

الفصل الثالث
مرحلة تنفيذ وإنتاج البرمجية

الفصل الرابع
مرحلة تجريب وتطوير البرمجية

الجزء الثاني
إنتاج برمجيات الوسائط المتعددة التفاعلية

الفصل الخامس
التعرف على بيئة فيجوال بيسك

الفصل الخامس
التعرف على بيئة فيجوال بيسك

الفصل السادس
تصميم وإنتاج شاشات النصوص والعرض

(تابع) الفصل السادس
تصميم وإنتاج شاشات النصوص والعرض

الفصل السابع
تصميم وإنتاج شاشة القائمة الرئيسية والقوائم الفرعية

الفصل الثامن
تصميم وإنتاج شاشات التمارين والاختبارات

الفصل التاسع
التعامل مع الصور وتحريكها ، والصور المتحركة ولقطات الفيديو

الفصل العاشر
طرق الانتقال من شاشة إلى أخرى

الجزء الأول

دورة إنتاج
البرمجية التعليمية

الفصل الأول

مرحلة تصميم وإعداد البرمجية

الفصل الثاني

مرحلة كتابة سيناريو البرمجية

الفصل الثالث

مرحلة تنفيذ وإنتاج البرمجية

الفصل الرابع

مرحلة تجريب وتطوير البرمجية

تمهيد

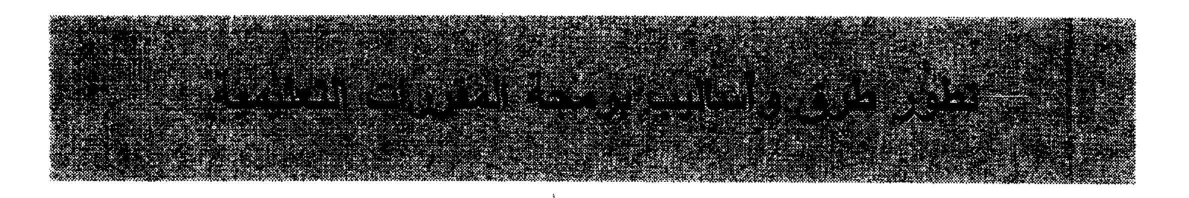
لقد كانت العقبة أمام من يقومون بإعداد المقررات الموكلون بتدريسها في صورة برمجيات تعليمية تكمن في ضرورة إلمامهم بمعرفة واسعة عن كيفية برمجة الحاسوب ، مما شتت تركيزهم بين النواحي التربوية والنواحي الفنية ، وقد كانت هناك عدة حلول لهذه العقبة والتي انتهت بتطوير حزم وبيئات جديدة تعرف بنظم التأليف *Authoring System* وأخرى متميزة تعرف بحزم تأليف برمجيات الوسائط المتعددة *Multimedia Authoring System* ؛ والتي صممت خصيصا للمعلمين لإنتاج البرمجيات التعليمية ، **وهي من السهولة بمكان حيث أن استخدامها لا يتطلب من المعلم أي خبرة في البرمجة .**

ومع الانتشار الهائل للحواسيب الشخصية ، والانخفاض الحاد في كلفتها وسهولة استخدامها ، ومع ازدياد حجم المادة التعليمية وتعقيد محتوياتها بصورة أصبح معها الكتاب المطبوع وما يصاحبه من وسائل تقليدية عاجزا عن تقديمها بشكل فعال . **أدرك المعلمون والمتعلمون والأباء ومطوري البرامج التعليمية والمناهج الإمكانات الهائلة التي يتيحها الحاسوب لإثراء العملية التعليمية ، لذا فهو ينتشر بسرعة فائقة في الفصول والمنازل والمدارس والمكاتب ومراكز التدريب كأداة فعالة لتقديم المادة التعليمية.**

بالرغم من الإمكانات الهائلة للحاسوب في عمليتي التعليم والتعلم فما زالت البرمجيات التعليمية بعيدة كل البعد عن **مشاركة المعلمين العرب في إنتاجها والتخطيط لها** ، علي الرغم من اعتقاد العديد من التربويين أن المعلمين – وهم بلا منازع – أقدر الأشخاص للقيام بذلك من حيث خبرتهم في انتقاء وعرض المادة التعليمية ، **وقد أثبتت التجارب أن معظم المعلمين يمكن أن يصبحوا مؤلفين متميزين للبرمجيات التعليمية .**

وقد لاح في الأفق بوادر عصر ذهبي لتطوير عمليتي التعليم والتعلم بشكل جذري، فقد وجد المعلم وتلميذه في الحاسوب غايتهما المنشودة ، وهو أداة المدرسة العربية والمعلم العربي والطالب العربي للحاق بعصر

المعلومات مع الحفاظ على الهوية العربية المتمثلة في استخدام اللغة العربية كتابة ولغة تعلم، بل ولغة برمجة .



قديما وفي بداية استخدامات الحاسوب في عمليتي التعليم والتعلم ؛ كانت تعد البرمجيات التعليمية المتواضعة باستخدام **لغات البرمجة العامة** *General Purpose Languages* مثل لغة فورتران *FORTRAN* ولغة بيسك *BASIC* ولغة بسكال *PASCAL* ولغة سي *C Language* ، والتي يتطلب استخدامها إلى خبرات واسعة وإتقان مستخدميها لأساسياتها والبرمجة بها ؛ الأمر الذي يصعب بل يستحيل توافرها في معلمي المواد ، كما تفتقر تلك اللغات نفسها إلى الوسائل العملية لتصميم الأساليب المتنوعة لتقديم المادة التعليمية ، مما أدى إلى إهدار الكثير من الجهد والوقت في سبيل إنتاج بعض الدروس المبرمجة بها ، إضافة إلى البطيء الشديد عند تنفيذ الدروس المنتجة بتلك اللغات.

وبعد فترة من الزمن ظهرت **لغات تأليف خاصة** *Authoring Languages* بها بعض الإمكانات التي تتيح للمستخدم تصميم الدروس التعليمية مثل لغة *Plato* ولغة بايلوت *PILOT* ولغة أيه. بي .إل *APL* ولغة كاتب العربية – وهى اللغة التي وضع أساسياتها الدكتور / محمد مندورة ورفاقه بالمملكة العربية السعودية – وأخيرا ظهور ما يسمى *Visual-C* ، *Visual-Pascal* والتي تعمل في بيئة ويندوز *Windows* ، ورغم مزايا تلك اللغات عن لغات البرمجة العامة لم تعفى المؤلف من مهمة البرمجة لكي يركز على الجانب التعليمي والتربوي.

ثم ظهرت **نظم تأليف المقررات** *Course Authoring Systems* ، وهي التي لا تحتاج من المؤلف أو المعلم أي خبرة في البرمجة ، حيث يقوم النظام بمساعدة المؤلف علي إنتاج البرمجية التعليمية بالمواصفات التي يضعها ، ليركز المؤلف على دقة انتقاء المادة التعليمية وأسلوب عرضها وتنظيمها وكيفية تحقيق الأهداف التعليمية وتلبية الغايات التربوية ، ومن أمثلة

هذه النظـــم : *Mac Prof. Authoring , Talk Tutor , The Educator ,* *Authority* وبالطبع يستفيد المتحدثون باللغة الإنجليزية من هذه الأنظمة إلـــى حد كبير . وباللغة العربية يوجد نظام **صاد** (صخر لإعداد الدروس) ؛ وهـــى المبادرة الخلابة التي قام بها الدكتور نبيل على وفريق التطويـــــر بالشـــركة العالمية للإلكترونيات.

وأخيرا ظهر ما يسمى **بأنظمة تـــأليف برمجيـــات الوســـائط المتعـــددة** *Multimedia Authoring Systems* والتي تتيح لمنتيجي المقررات الدراســـية المبرمجة الاستفادة من الوسائط التعليمية المختلفة كالفيديو وأجـــهزة تشـــغيل أقراص الليزر CD-ROM ، والأصوات والصور الثابتة والمتحركة ، ودمـــج العديد من المشاهد والمواقف بالبرمجيات المستهدفة، هذا وتتميز هذه الأنظمـــة عن سابقتها بأنها مزودة بإستراتيجيات واختيارات وأدوات خاصة للتصميـــــم التعليمي ، الأمر الذي يسمح بمرونة كبيرة أثناء مرحلة التنفيذ تمكن المؤلفيـــن والمنفذون من إعداد الدروس وفقا للمعايير التربوية الســـليمة ، دون خـــبرة كبيرة في برمجة الحاسوب .

تقدم نظم التأليف الجديدة أدوات قوية ومرنة لتحويل المحتويات الجـــامدة إلى تطبيقات تفاعلية . وتتباين هذه النظم فيما بينها بالنسبة للمواصفات التقنية والفنية ، لذا فعلي مستخدمها أن يختار النظام الأنسب لمشــــروعه ومســـتوى خبرته . فنظام فيجوال بيسك *Visual Basic 4.0* علي سبيل المثال يقدم أدوات تعامل مع قواعد البيانات وتحكم عن طريـــق البرمجـــة، لكـــن اســـتخداماته مقصورة علي أجهزة IBM والأجهزة المتوافقة معها فلا يمكـــن اســـتخدامه مثلا مع أجهزة أبل ماكينتوش ، هذا ويتوازن نظام دايركتـــور *Director 5.0* بين القوة والملائمة ، لكنه يحتاج من مستخدميه إلى إتقان لغة لينجـــو *Lingo* لأداء المهام المعقدة.

تقع أنظمة تأليف الوسائط المتعددة *Multimedia* في منتصف المسافة بين برامج العروض التقديمية بالوسائط المتعددة *Multimedia Presntation* مـــن طرف ، وبين لغات البرمجة المتقدمة المستخدمة لتطوير تطبيقات الوســـائط المتعددة التجارية (مثل الألعاب والموساعات العلمية) من الطرف الآخر. تهتم نظم تأليف الوسائط المتعددة بإنشاء تطبيقات شديدة التفـــاعل ، تتدفـــق فيـــها المعلومات بالإتجاهين : من التطبيق الي المتعلم ، ومن المتعلم الي التطبيـــق ،

أما برامج العروض التقديمية ، فتهتم بشكل رئيسي بنقل المعلومات بإتجاه واحد من التطبيق الى المتعلم ، ويقتصر تفاعل المتعلم علي التجوال في هذه التطبيقات.

نضجت نظم تأليف الوسائط المتعددة بالمقارنة عما كانت عليه في السابق، وبرز جيل جديد منها ليغير من صورة الحواسيب الشخصية ، فقد كانت هذه النظم تحتاج في السابق الى مهارة المحترفين لإستخدامها ، اما اليوم ، فإن انشاء تطبيقات تفاعلية صوتية وفيديوية عالية الجودة ، لا يتطلب الكثير من الخبرة ، ويستطيع اشخاص عاديون تطوير تطبيقات وسائط متعددة جيدة. حيث يمكن تقسيم نظم تأليف برمجيات الوسائط المتعددة الى مجموعتين: مجموعة اعتمدت بالكامل علي تقنية وجه وانقر *point-and-click* وإحتاجت الأخرى الى كتابة نصوص برمجية بسيطة *scripting*

هذا ولم تعد الأصوات ولقطات الفيديو اوساطا جامدة غير ديناميكية فخضعت تقنيات الفيديو للعديد من التطوير ، فقد اصبحت تقنية MPEG – (MPEG : *Motion Pictures Experts Group* نظام فيديو رقمي طورته مجموعة خبراء الصور المتحركة) – تنتج لقطات الفيديو بالقياس الكامل للشاشة ، وبمعدل ٣٠ اطارا في الثانية . ويعتبر هذا تطورا كبيرا بالمقارنة مع خرج ملفات *AVI* التي لا يتجاوز قياس نافذة العرض فيها عن قياس طابع البريد. ان معظم الأنظمة التالية يمكنها التحكم ببطاقات اعادة عرض من خلال اوامر واجهات التحكم بالوسائط *MCI* (MCI : *Media Control Interface*)

تضمنت مهمة التأليف علي شبكة ويب *Web authoring* إنشاء تطبيقات ، يمكن تشغيلها عن طريق تطبيق مساعد *Helper application* من داخل متصفحات *browsers* الشبكة العالمية انترنيت ، أو انشاء تطبيق معقد مستقل ، مصمم لدفق بيانات مزود الشبكة عبر وصلات *TCPlIP* ، ومنها الى المتصفح الخاص بالمستخدم بدون ان يحتاج الى انزال التطبيق المعني من المزود.

ان الثورات التي تحدث الآن في تقنيات تأليف الوسائط المتعددة ، تمثل تغييرا جذريا في التطبيقات التي سننشئها ، فالإثارة التفاعلية ، ومغامرات الرسومات ثلاثية الأبعاد عبر هئية VRML (*Virtual Reality Modelin Language*)

لأفضل برامج العاب الحاسوب في هذه الأيام سوف تنسج طريقها الينا لتصبح تحت سيطرتنا الشخصية ، وتمكننا من انتاج كتالوجات فورية تشبه الألعاب .

تتكون مشروعات البرمجيات عادة من شاشات مستقلة تعمل كحاويات لكائنات الوسائط المتعددة . أدوات التحكم الرسومية تساعدك علي التحرك بين الشاشات بحرية تامة ، أما أدوات التحكم الأخرى والبرامج فتسمح لك بربط تصرفات المتعلم بالأهداف الموجودة علي الشاشات . هذا وتعتبر مساندة التعامل مع الملفات ذات أهمية كبرى للتعامل مع الأشكال المتنوعة من ملفات الأديو والفيديو والرسوم ولدمج هذه الوسائط داخل البرمجيات . بعد جلب الأهداف يمكنك إضافة بعض الخواص التفاعلية بتغيير خواص الأهداف .

أهم نظم تأليف برمجيات الوسائط المتعددة

تقدم جميع النظم التالية واجهة رسومية جيدة للتحكم في مواضع الوسائط، بالإضافة إلى أدوات تصميم لإضافة الأزرار والنصوص والقوائم وأنواع الأهداف *Objective* المختلفة . تتضمن تلك النظم علي لغات برمجة قوية تسمح للمنفذ بالتحكم الدقيق ، والقيام بالحسابات ، وقبول ما يدخله المتعلم والتعامل معه ، بل والربط ببرامج أخري.

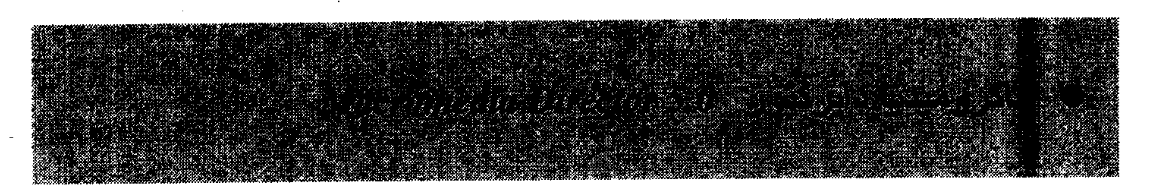
وسوف نستعرض فيما يلي بإيجاز أهم هذه النظم ، والتي تعمل في بيئة ويندوز ، والتي تعتبر نظم تطوير برمجيات وسائط متعددة ، مكتملة النمو مناسبة لإنتاج تطبيقات مستقلة بذاتها . تعتمد جميع هذه النظم علي مايسمى بتوجيه الأهداف ، حيث تتيح بسحب واسقاط الوسائط المختلفة (الأهداف) داخل شاشات البرمجية ، هذا ويمكن انجاز العمل من خلالها دون الحاجة الى البرمجة ، مع الإحتفاظ بخيار كتابة الأكواد للحصول علي تطبيقات متقدمة.

• مثلث مبدئيا تول بوك *Asymetrix Multimedia Toolbook 4.0*

بدأ نظام **تول بوك** كنظام تأليف محدود الإمكانات ، ثم بلغ الرشـــد الآن بإصداره الرابع ، الذي يقدم توازنا جيدا بين القوة وقابلية الاســتخدام ، ممــا يجعله اختيارا ممتازا للمؤلفين الجدد . يعتمد النظام علي كتـــاب مجـازي : الصفحات عليها أزرار وحقول بيانات ، ووسائط متعددة ، يعيــب البرنـــامج الحاجة إلى القص واللصق لنسخ الصفحات.

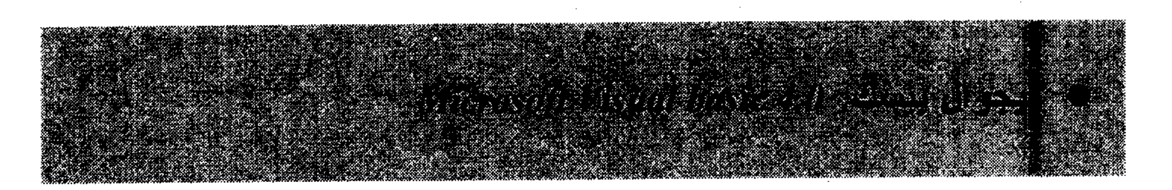
عند بداية مشروع البرمجية يسألك **تول بوك** سلسلة من الأســئلة ليبنـــي إطار عمل لمشروع البرمجية وفقا للمواصفات التي تضعها. هــــذا وتســمح صناديق الخواص *Properties Boxs* بتغيير خواص الأهداف *Objective* ممــا يريحك من كتابة برامج لمهام الملاحظة والتفاعل ، بينمـــا توفــر مســاطر الأدوات *tool bar* وسيلة سريعة للأوامر الشـــائعة ، والعديــد مــن تحريــر ومعالجة الوسائط *Media Editors* بالإضافة إلى كتابة الجمل والأنشطة عــن طريق ما يسمى *OpenScript* . قد تحتاج إلى استخدام لغة البرمجة بنظام **تول بوك** لتنفيذ بعض الوظائف التي لا تتيحها لوحة الأدوات مثل البحث عن نص مثلا ، ولكن وجود مكتبة كبيرة من الروتينات الفرعية *Subroutine* الجــاهزة للصق تسهل عليك المهمة. يتضمن نظام **تول بوك** العديد مــن أدوات ربــط ودمج الأهداف ***Object Linkinmg & Embeding OLE*** المطورة مـــن قبــل المستخدم مفيدة في التعامل مع قواعد البيانات والاحتياجات الخاصة ، ولكــن دون نظام فيجوال بيسك في ذلك.

برنامج التركيب في بيئة ويندوز الموجود ضمن **تول بوك** يقوم بتركيب المنتج النهائي والمحتويات ، ويمكن ضبطه ليناسب العمل من خلال مشغلات سي دي روم CD-ROM ، هذا بالإضافة إلى توافـــر إمكانيـــة أن تتواصــل منتجات **تول بوك** مع شبكة انترنيت.



لقد تطور نظام **دايركتور** الشائع الاستخدام أخيرا من خــلال الإصــدار الخامس ، والذي يدعم بيئة الــ ٣٢ بت ، حيث يقدم مرونة ودقة ملموستان ، فقد تغلب علي بعض نقاط الضعف بالإصدارات الســابقة ؛ بمــا فــي ذلــك المساندة الكبرى للملفات والتعامل مع النصوص ، وقابلية البرمجة ، إضافــة إلى توفير ما يسمى بالمسرح أثناء التأليف ، والتي تعتبر خاصية فريدة بنظام **دايركتور** . يسمح النظام بالتحكم في الرسوم المتحركة إطارا إطارا ، كمــا يتيح بتشغيل قناتي أديو وفيدو في وقت واحد ، إن وجود واجهات التحويــلات *Transformations* التي تشبه الجداول الإلكترونية تسهل التحكم فــي مــهام وتفاعلات الأهداف بدقة فائقة . بجانب البرمجــة تتوافــر إمكانــات تتيــح للمطورين استبدال المحتويات بكل سهولة ، حيث يحتوى النظام علي حوالــي ١٠٠ أمر جديد من أوامر لغة لينجو للتحكم بهذه الإمكانات .

يعتبر موطن القوة بنظام **دايركتور** هو مساندته لبيئات عمل كثيرة حيــث يسمح **دايركتور** بتأليف التطبيقات علي أجهزة مــاكينتوش أو أجــهزة IBM والأجهزة المتوافقة معها من خلال بيئات ويندوز ويونكــس ، إضافــة إلــى إمكانية تقديم التطبيقات المنتجة من خلال هذا النظام علي WWW (*Word Wide Web*) لشبكة انترنيت بكل سهولة. هذا وتعتبر الــبرامج الإضافيــة المســاندة لدايركتور مثل *Extreme 3D* ، *xRes 2.0* : لتعديل وطباعة وإعداد الصــور عالية الدقة ، وكذا *Sound Forge XP* نقاط قوة أخرى بلا شك.



يتمتع **فيجوال بيسك** بشعبية كبيرة بين منتيجو البرمجيــات التعليميــة ، لقوته وسهولة استخدامه نسبيا إذ انه يقدم الكثير من الأدوات لبناء تطبيقــات الوسائط المتعددة ، ويعيب عليه انه يعمل علي بيئة ويندوز فقط .

بيئة العمل في **فيجوال بيسك** مسطرة الأدوات *tool bar* وصندوق الأدوات *tool box* ، ونافذة المشروع *Project Windows* ؛ حيث تتيح لنا أدوات التحكم المختلفة بإضافة والرسوم والأصوات ولقطات الفيديو الرقمي ، علي نماذج المشروع *Project Forms* ، حيث تحدد سلوك هذه الأهداف في نوافذ الكود *Code Windows* ، والتي عن طريق قوائم الخواص *Properties Lists* يمكن تغيير خواص تلك الأهداف. يحتوى معالج الأكواد *Code Editor* علي قائمة أوامر مفيدة من الأوامر والأحداث *events* لمساعدة المستخدمين . إضافة إلى احتواء النظام علي عدد لا بأس به من اختيارات إنشاء أو تعديل عناصر الوسائط المتعددة . هذا ويقبل **فيجوال بيسك** العديد من أنواع الرسوم والصور ، إضافة إلى أدوات تحكم *OCX* الموجودة ضمن النظام التي تقبل المزيد من تلك الصور والرسوم.

ضمن اختيارات الوسائط المتعددة بالنظام يمكنك التحكم في أجهزة *MCI* وتشغيل ملفات الأديو في صيغة *wav* ، وعرض لقطات الفيديو في صيغة *avi* والتحكم في لوحات الأديو في صيغة *MIDI* . هذا وتتيح أدوات **فيجوال البيسك** القوية امكانية التعامل مع قواعد البيانات الخارجية لتتيح إنشاء قواعد بيانات داخل البرمجيات عن طريق محرك قواعد البيانات *JET* ، وهو نفسه المستخدم في ميكروسوفت أكسس *Microsoft Access*

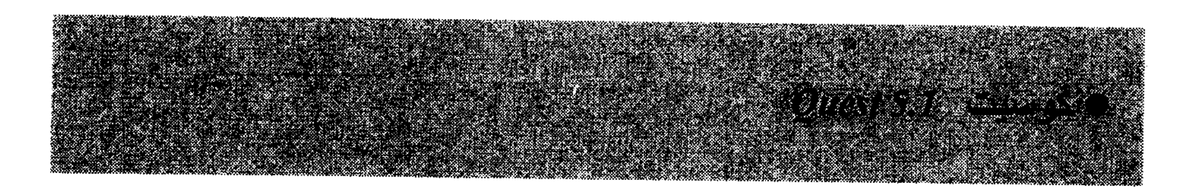
• أوراكل ميديا أوبجيكتس *Oracle Media Objects 1.0*

أولئك المعتادون علي أداة التأليف ماك هايبركارد *Mac HyperCaerd* سيشعرون بالألفة مع نظام **أوراكل ميديا أوبجيكتس** ، حيث يعمل النظام من خلال عدد من الحزم التي تحتوي علي عدد من الكروت ، وعلي الكروت توجد طبقات من الرسوم وأهداف الوسائط المتعددة ، يتم ربطها وإدارتها بلغة أوراكل ميديا توك *Oracle Media Talk* ، بينما تساعد الأدوات المتوفرة بالنظام علي التعامل مع قواعد البيانات الخاصة بأوراكل *Oracle Media Servers* لتقديم المحتويات من خلال شبكة البيانات.

إنشاء البرمجيات من خلال هذا النظام ليس صعبا ، حيث يتم التنقل بين شاشات البرمجية بسهولة . تستخدم مسطرة الأدوات لإنجاز المهام المعتادة ، كما تستخدم صناديق الحوار *Dialog Boxs* لمشاهدة وتعديل خواص الأهداف *Objects properties* ، هذا وتتيح نافذة الاختيار التحكم الدقيق في تلك الأهداف. كما تتيح نافذة اختيار الكود بالتحكم في المتغيرات *variables* مما يوفر علي المستخدم عناء البحث عن الأخطاء المنطقية ، إضافة إلى توافر البرنامج المساعد *DataSheets* الذي يتيح للمستخدم عرض الجداول والأرقام والنصوص ومسارات تحريك الأهداف.

• ميديا فورج *Strata Media Forge 2.0*

من السهل استخدام هذا النظام الذي يعتمد علي توجيه الأهداف *Object Oriented* كغيره من الأنظمة السابقة ، حيث يسمح بسحب وإسقاط الوسائط داخل شاشات البرمجية ، ويمكن إنجاز العمل دون الحاجة إلى البرمجة ، مع الاحتفاظ بخيار كتابة الأكواد للحصول علي تطبيقات متقدمة . لنظام **ميديا فورج** بناء طبقي يبدأ بالشاشة الرئيسية للبرمجية ثم يتفرع إلى الشاشات التالية، حيث يوفر مسرحا يمكن تغيير حجمه لضم وتصميم المحتويات إلى جانب صندوق أدوات ممتلئ بأدوات تضمين الأهداف القوية مثل أدوات الربط والدمج *OLE* وأوامر *MCI* ونصوص *RTF* وعناصر الرسوم . تتضمن أغلبية الأدوات علي خواص مصاحبة تساعد علي تصاميم الانتقال والقفز والمقارنة والانتظار ، لذا لا يحتاج المستخدم إلى كتابة أية اكواد بلغة ميديا بيسك *MediaBasic* المعتمدة داخل نظام **ميديا فورج** ، هذا ويمكن للمستخدم تعديل خواص الأهداف باستخدام قوائم الخواص ، كما توجد اختيارات خاصة بالمتغيرات *variables* وأدوات لإدخال البيانات تعرف باسم أقنعة إدخال البيانات *data entry masks* . يعتبر **ميديا فورج** متأخرا عن بقية الأنظمة فيما يتعلق بدقة الوسائط المتعددة التي يقدمها ، والتحكم في تزامن تشغيل الأديو مع الفيديو ، بالرغم من احتوائه علي إمكانية النشر عبر شبكة انترنيت.



● كويست *Quest 5.1*

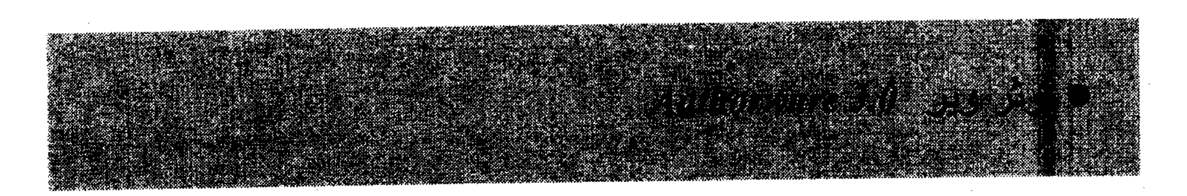
يتألف نظام *Quest 5.1* من تطبيقين صغيرين مرتبطين مع بعضهما البعض : التطبيق الأول هو عبارة عن مخطط نهجي للأيقونات ، يحدد جميع الارتباطات بين الإطارات . والتطبيق الثاني عبارة عن محرر إطارات يستخدم لإنتاج الرسومات والبرمجة ، لكل إطار من الإطارات ، حيث يمكن المستخدم من اختيار وترتيب تعليمات أيقونة الشكل وبسيطة التشعب ، إلا أن القيام بأي عمل آخر اكثر تعقيدا ، يتطلب استعمال لغة *Quest C* المشابهة إلى حد ما للغة سي.

● أيكن أوثر *IconAuthor 7.0*

تستحق بيئة تطوير التطبيقات في نظام *IconAuthor 7.0* التزكية ، ويتضمن النظام مخططا نهجيا مجازيا يمكن المستخدم من بناء تطبيق وسائط متعددة بدون الحاجة الى كتابة تعليمات برمجة ، وتفوق علي منافسيه في مجال الوصول الي البيانات ، ومعالجتها ، اضافة الى انه يحتوى علي مجموعة من التي تمكن المستخدم من انشاء تطبيقات قوية .

حافظ نظام *IconAuthor* علي اعتماده علي الأيقونات ، الا انه استطاع عن طريق محرره *SmartObject Editor* الذي اضحى يتحكم بالأهداف التي لها اهمية خاصة في انشاء تطبيقات الوسائط المتعددة ، حيث يقوم هذا المحرر يجمع الأهداف المختلفة كالصور ، ولقطات الفيديو ، والرسوم المتحركة ، والنصوص ، والأزرار ، وصناديق القوائم ... الخ ضمن ايقونات جاهزة للعمل في صورة مشروع متكامل ، اضافة الي ما يتمتع به النظام من دعم لقواعد البيانات عن طريق *ODBC*.(*Open Data Base Commectivity* *support* وهو عبارة عن دعم الإرتباط المفتوح لقواعد البيانات ، والذي يزودنا الوصول الى واجهة برمجة قواعد البيانات . وهي تقنية طورتها ميكروسوفت بلغة مشتركة

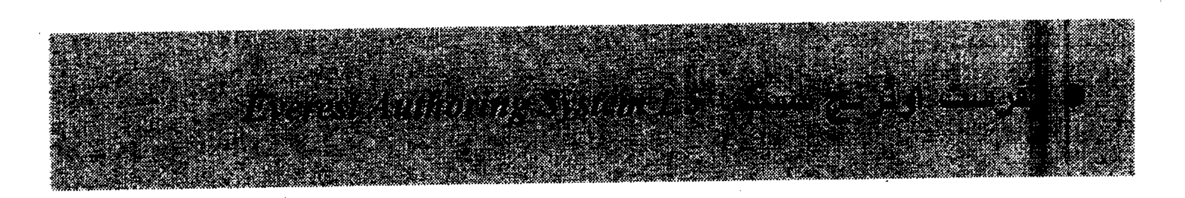
لتطبيقات ويندوز ، يمكنها التعامل مع مختلف قواعد البيانات المخزنة علي اقراص محلية او علي اقراص الشبكة).



يساعد نظام *Authorware 3.0* بفضل واجهة استخدامه الفريدة من نوعها، المستخدمين الذين ليس لديهم خبرة في البرمجة ، علي بناء تطبيقات معقدة بسهولة ويسر ، وبدون أن يحتاجوا الي كتابة سطر واحد من البرمجة، مما يجعلة نظاما قويا نسبيا.

يتميز هذا النظام بأنه يسمح لعد اشخاص بالإشتراك معا في تطوير تطبيق ما . فيستطيع الفنانون ، مثلا ، تحميل مكتبة مع وسائط مختلفة ، ويستطيع المبرمجون تأمين القوالب اللازمة للتفاعلات المعقدة ، ويستطيع مصمموا واجهات الإستخدام أن يدمجوا هذه الأهداف في التطبيق النهائي ، أضافة انه يمكن لمصمم واحد لاخبرة له في مجال البرمجة ، أن يقوم وحدة بالعمل كله.

تمكنا بعض الأيقونات بهذا النظام من التحكم بالوسط المطلوب علي الشاشة ، كالعرض والمسح .. إلخ .



يعتبر نظام *Everset Authoring System 1.5* أداة سهلة الإستخدام ، ويحتاج الى القليل من الخبرة البرمجية التقليدية أو حتى بدون خبرة برمجية ، لإنشاء تطبيقات متطورة .

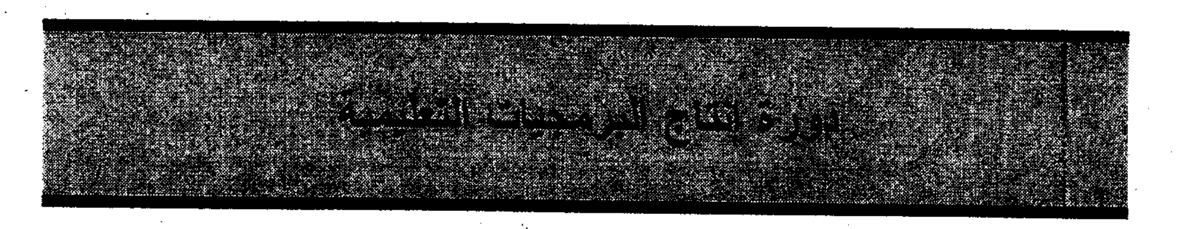
ولكي تعم الفائدة ويتمكن المعنيين من الاختيار السليم بين الأنظمة السابقة نورد فيم يلي **جدول مواصفات مقارن لتلك الأنظمة** :

جدول (١)
جدول مواصفات مقارن للأنظمة

	تول بوك ToolBook 4.0	دايركتور Director 5.0	فيجوال بيسك Visual Basic 4.0
منصة العمل *Authoring Platform*	Windows 3.1 , 95 & Windows NT	Mac & Windows 3.1 , 95 & Windows NT	Windows 3.1 , 95 & Windows NT
لغة البرمجة *Scripting Language*	OpenScript	Lingo	Visual Basic
الرسوم *Graphics*	PICT, BMP DIB ,WMF CGM, PIC DRF, GIF ,PXC, PIC, TIFF, CDR EPS, TGA PhotoCD	PICT, BMP DIB ,WMF CGM, PIC DRF, GIF, PXC, PIC, TIFF, CDR EPS, TGA PhotoCD	PICT, BMP DIB ,WMF CGM, PIC DRF, GIF, PXC, PIC TIFF, CDR EPS, TGA PhotoCD
هيئات ملفات النصوص *Text*	ASCII, RTF, TXT ● إضافة الظلال ● ابراز النص	ASCII, RTF, TXT	ASCII, RTF, TXT ● إضافة الظلال ● ابراز النص
هيئات ملفات الأصوات *Audio*	CD Audio MIDI, WAV,SND ● تكرار الصوت ● ايقاف الصوت	CD Audio MIDI, WAV ● تكرار الصوت ● ايقاف الصوت	CD Audio MIDI, WAV ● تكرار الصوت ● ايقاف الصوت
هيئات ملفات الرسوم المتحركة *Animation*	FLC, FLI • بالتعاقب	FLC, FLI PICS • بالأشباح	FLC, FLI • بالتعاقب
هيئات ملفات الفيديو *Vidou*	AVI Quick Time	AVI , MPEG Quick Time ActiveMove	AVI
دعم قواعد البيانات *ODBC*	OLE controls MediaBasic	OLE controls	OLE controls
الإنترنيت *Internet Tools*	GIGýF HTML	GIGýF HTML	**لاتوجد**

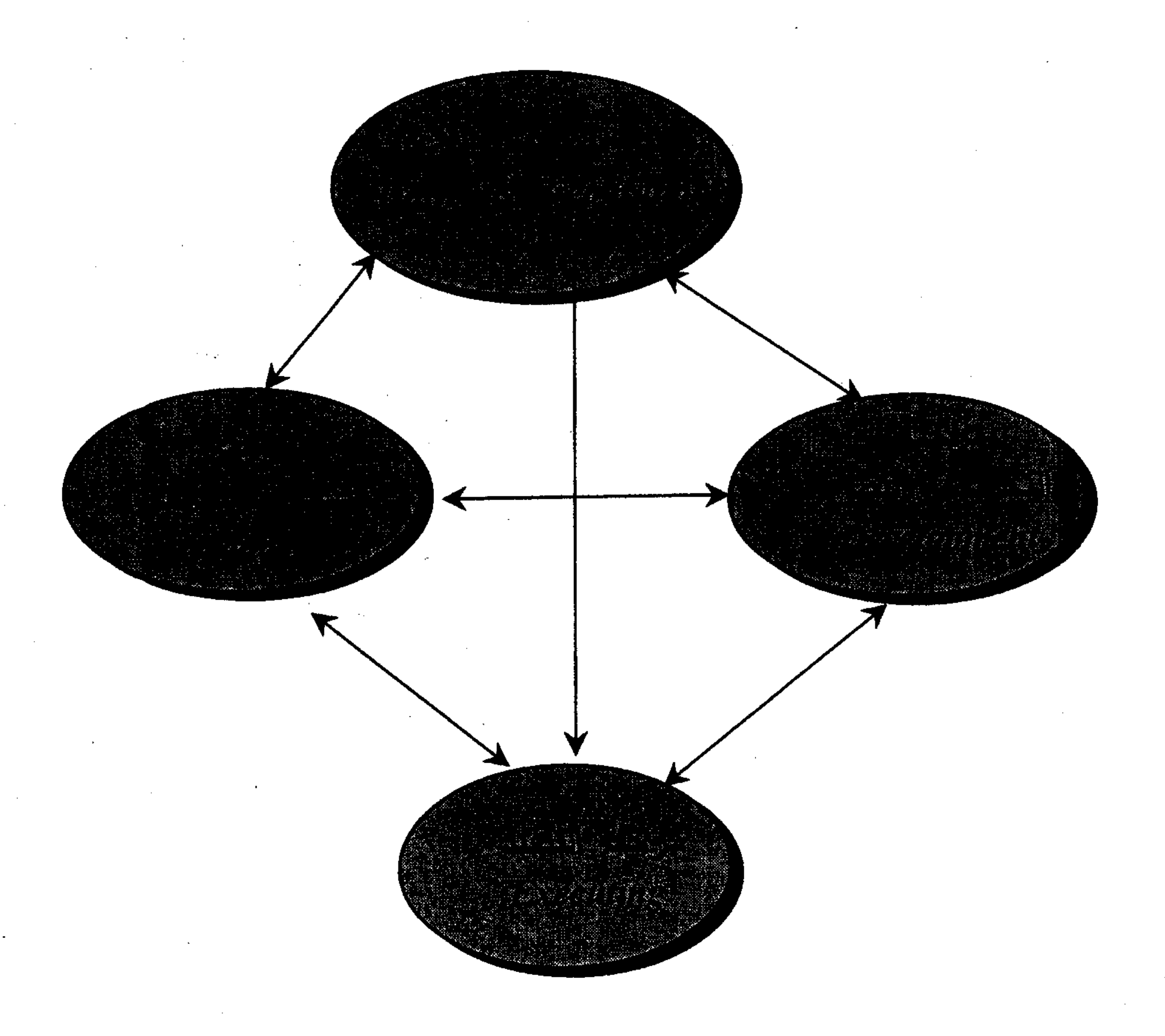
(تابع) جدول (١)
جدول مواصفات مقارن للأنظمة

[illegible] Quest 5.1	[illegible] Strat Media Forge 2.0	[illegible] Oracle Media Objects 1.0	
Windows 3.1 , 95 & Windows NT	Windows 3.1 , 95 & Windows NT	Windows 3.1 , 95 & Windows NT	**منصة العمل** *Authoring Platform*
Quest C	MediaBasic		**لغة التعامل** *Scripting Language*
PICT	BMP ,WMF PIC, PXC, PIC TIFF, EPS, TGA, JEPG	PICT	**الرسوم** *Graphics*
ASCII	ASCII, RTF	ASCII	**هيئات ملفات النصوص** *Text*
MIDI, WAV	CD Audio MIDI, WAV	MIDI, WAV	**هيئات ملفات الأصوات** *Audio*
لا توجد	FLC, FLI • بالأشباح	هيئات خاصة • بالتعاقب	**هيئات ملفات الرسوم المتحركة** *Animation*
AVI	AVI , MOV Quick Time & Intel Indo	Quick Time	**هيئات ملفات الفيديو** ***Vidou***
OLE controls	OLE controls MediaBasic	Oracle7	**دعم قواعد البيانات** ***ODBC***
لا توجد	GIGýF HTML	**لا توجد**	**الإنترنيت** *Internet Tools*

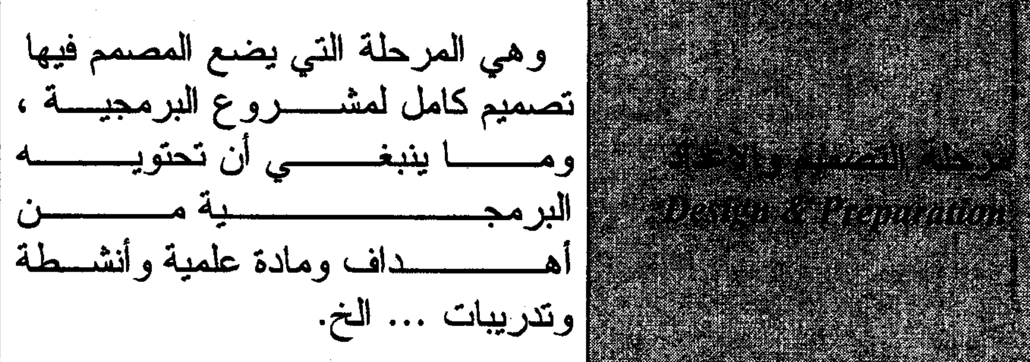


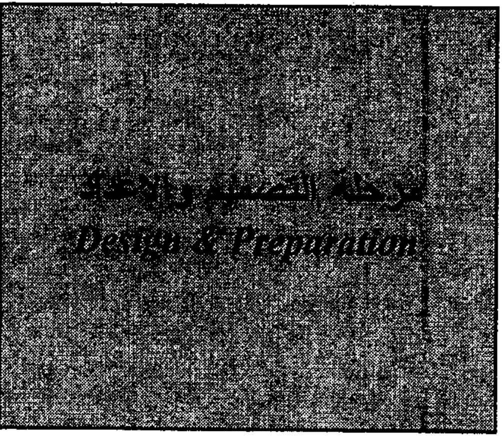
تتكون البرمجية التعليمية عادة من عدة موضوعات ، حيث يتكون الموضوع بدوره من عدة دروس ، ويتكون كل درس من عدة فقرات ، وتتكون الفقرة من عدة نوافذ أو شاشات تعرض من خلالها المواد التعليمية : في صورة تدريس خصوصي *Tutorial* والذي عادة ما يتضمن العرض *Presentation* مدعما بالصور الثابتة *images* والرسوم والصور المتحركة كالقطات الكرتون ولقطات الفيديو *Video Clip* المؤثرات الصوتية *Sound* والحركة *Animation* والحوار *dialog* وعرض أمثلة وتمارين *Examples & Exercises* ، وتقديم مفردات اختبار *Test Items* : تشخيصية *Diagnostic* أو بنائية *Formative* أو نهائية *Summative* أو إتقان *Mastery* ، إضافة إلى مجموعة ملفات لحفظ أداء المتعلمين.

تمر عملية إعداد البرمجيات التعليمية بعدة مراحل ، قبل أن تخرج بالشكل النهائي الذي تعرض به ، وقد يقوم بهذه العملية مجموعة مختلفة من الأفراد أو المعلمين ينبغي أن تتوافر لديهم خبرات ذات مواصفات محددة ، **وتمر عملية إنتاج البرمجية التعليمية عادة بأربعة مراحل تعرف بدورة إنتاج البرمجية والشكل رقم (١) التالي يوضح هذه الدورة :**



شكل (١)
يوضح دورة انتاج البرمجية التعليمية

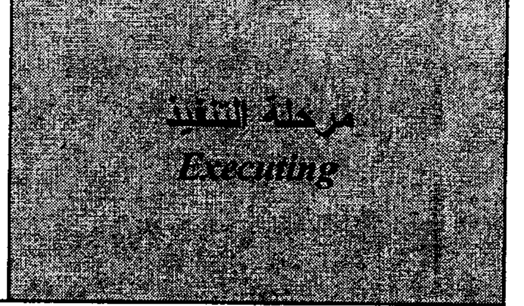




وهي المرحلة التي يضع المصمم فيها تصميم كامل لمشروع البرمجية ، وما ينبغي أن تحتويه البرمجية من أهداف ومادة علمية وأنشطة وتدريبات ... الخ.



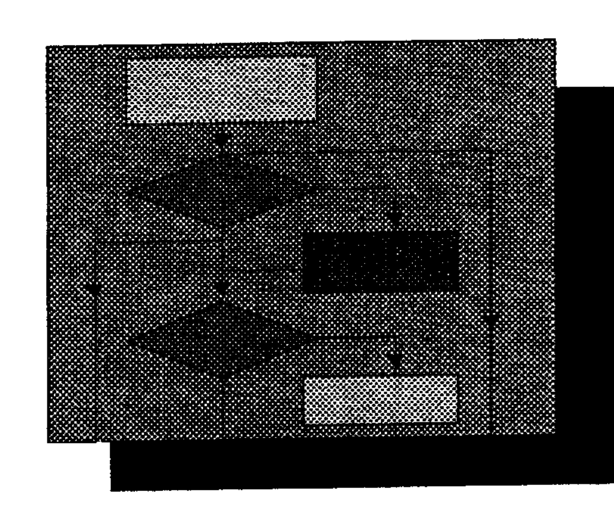
وهي المرحلة التي يتم فيها ترجمة الخطوط العريضة التي وضعها المصمم إلى إجراءات تفصيلية وأحداث ومواقف تعليمية حقيقية علي الورق مع وضع في الاعتبار ما تم إعداده وتجهيزه بمرحلة الإعداد من متطلبات .



وهي المرحلة التي يتم فيها تنفيذ السيناريو في صورة برمجية وسائط متعددة تفاعلية ، مع كتابة بعض البناءات المنطقية *Code* .



وهي المرحلة التي يتم فيها عرض البرمجية علي عدد من المحكمين المختلفين ؛ بهدف التحسين والتطوير.



الفصل الأول

مرحلة

تصميم وإعداد البرمجية

تمهيد

عادة ما يتولى **تصميم** وتخطيط البرمجية ***Design & Planing*** أفضل المعلمين خبرة في تخطيط المناهج التعليمية وتصميمها ، وأن يكون متمكنا من المادة التعليمية موضوع البرمجية ؛ حيث أنه يقوم بالدور الرئيسي في وضع الخطوط العريضة التي ينبغي أن تسير عليها البرمجية المستهدفة : **فيقوم بتحليل المادة التعليمية موضوع البرمجية ، ويقوم بتحديد الأهداف التعليمية العامة والخاصة وصياغتها في صورة أهداف سلوكية وتصنيفها تصنيفا مناسبا والعمل علي سلسلتها وترتيبها في صورة أهداف موديولية ، ويقوم ببناء الاختبارات اللازمة لتقويم أداء المتعلمين في كافة مراحل البرمجية ، وان يحدد محتويات ومواقع وتوقيتات دورات التشخيص والعلاج، ويضع تصميما عاما لخط سير تعليم وتعلم التلاميذ المستهدفين ، وهذا أشبه بخريطة عامة توضح علاقات مكونات البرمجية ، وكيفية تسكين الطلاب في المقرر ، ومتابعتهم أثناء الدراسة . وطريقة تقويمهم .**

ينبغي أن يكون مصمم ومخطط البرمجية لديه فكرة عامة عن **الإمكانيات المختلفة لأجهزة الحواسيب والكيفية التي يعمل بها نظام التأليف المقرر استخدامه في إنتاج البرمجية المستهدفة** ؛ حيث سيساعده ذلك في القيام بعملية تخطيط سليمة.

إضافة إلى ذلك ينبغي علي مصمم ومخطط البرمجية أن يكون ملما **بأنماط البرمجيات التعليمية ، ومراحل تطورها** ، وكذا **كيفية نقد وتقويم البرمجيات التعليمية.**

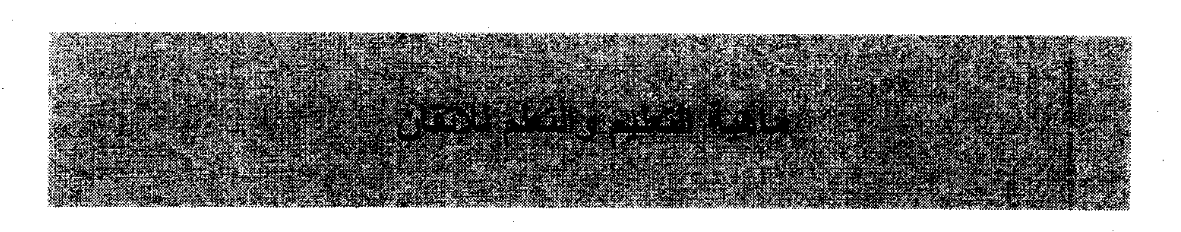
ومما هو جدير بالذكر هنا أن عملية تخطيط وتصميم البرمجيات التعليمية قد ارتبط إلى حد كبير بمفهوم **التعليم والتعلم للإتقان** ، إلى درجة أن أطلق البعض علي هذا الارتباط بالزواج الكاثوليكي الدائم الذي لا يوجد فيه انفصال؛ وبالتالي فإنه من **الأجدى عند استعراض خطوات تخطيط وتصميم**

البرمجية التعليمية ؛ استعراض ماهية التعليم والتعلم للإتقان ، وذلك من خلال ما يلي:

- ماهية التعليم والتعلم للإتقان .
- مفهوم التعليم والتعلم للإتقان .
- نموذج التعليم التعلم للإتقان.
- مراحل الإعداد للتعليم والتعلم للإتقان :

١- مرحلة تحليل وتنظيم المادة التعليمية.
٢- مرحلة صياغة الأهداف التعليمية وتصنيفها .
٣- مرحلة بناء الاختبارات وتحديد مستوى الإتقان.
٤- مرحلة إعداد أدوات ووسائل التقدير القبلي.
٥- مرحلة تحديد متطلبات التعليم القبلية .
٦- مرحلة التخطيط للوصول إلى الإتقان ؛ وهذا يتطلب ما يلي :

- اختيار طرق التعليم وإعداد الأنشطة والمواد التعليمية.
- إعداد خطة عرض المحتوى الوقائية
- إعداد خطة العرض العلاجية
- إعداد خطة العرض الإثرائية



إن لكل شخص **صفاته الفردية** التي لا يتساوى فيها مع أحد آخر والتي تنعكس **على سلوكه وتفكيره وتفاعله مع عناصر البيئة المحيطة به** ، وهذه الصفات قد تكون **فسيولوجية** كالمظاهر الجسمية والعضوية الظاهرة ، كما قد تكون **نفسية** مثل سرعة الفهم ، والإدراك، وقوة الملاحظة ، وإدراك

التفاصيل، والقدرة على التذكر، ودرجة الانفعال. كما قد تكون هذه الصفات مرتبطة بشخصية الفرد مثل ميله إلى الانطواء أو الانبساط والميل إلى الاجتماع مع الناس أو العزلة والميل إلى السيطرة أو التبعية إلى غير ذلك من سمات الشخصية.

وهذه الفروق بين الناس إنما أوجدها الله سبحانه وتعالي لحكمة كبيرة وهي أن يهيأ كل فرد للقيام بعمل من الأعمال التي يحتاجها المجتمع. وقد فطن المربون إلى مبدأ الفروق الفردية من وقت طويل ، إضافة إلى إقرارهم بأن لكل فرد حاجات خاصة به لابد من إشباعها أثناء تعليمه حتى يمكنه أن يصل إلى أقصى نمو تؤهله له هذه الصفات التي تميزه عن غيره من الأفراد ، وقد جرت المحاولات المتكررة للبحث عن **أفضل الطرق لإشباع هذه الحاجات الفردية وشملت مجالات عدة منها أسلوب تعامل المربي مع تلاميذه وطرق توصيل المعلومات إليهم، كذلك شملت أيضا أنواع المعلومات والخبرات التي تشبع هذه الحاجات.**

إن الفروق الفردية التي بين الأفراد إنما هي فروق كمية فقط وليست فروقا نوعية بمعنى أن جميع السمات والقدرات والاستعدادات متوفرة في كل فرد وأن الفروق بين الأفراد تنحصر فقط في مقدار توفر السمة أو القدرة لدى كل منهم فنجد مثلا أن هناك فروق بين الأفراد في مستويات ذكائهم ، ولكن بالتأكيد لا يوجد فرد خاليا تماما من الذكاء. وهكذا بالنسبة لجميع السمات الأخرى وان امتلاك الأشخاص لأي سمة إنما يتمثل بمقياس متصل اتصالا تاما يسمى المقياس المتصل *Continuum* لا نستطيع أن نحدد له بداية أو نهاية تحديدا قاطعا كما لا يمكن تقسيمه إلى فترات منفصلة. وبين كل درجة وأخرى على هذا المقياس نجد عددا لا حصر له من الدرجات الأخرى لنفس السمة.

إن توزيع السمة للأفراد المختلفين على المقياس المتصل لا يقتصر فقط على السمات النفسية وإنما يمتد أيضا ليشمل جميع السمات والقدرات العضوية وقوة الحواس ، وغيرها من السمات الجسمية أو الذهنية أو الانفعالية. وبالتالي

فانه يصبح من الخطأ أن نطلق على شخص أنه ذكي أو أنه غبي، انطوائي أو انبساطي ، بل يجب أن نكون أكثر دقة في تحديد درجة الذكاء ودرجة الإنطوائية...وهكذا. وإذا تم قياس سمة معينة سواء كانت جسمية أو نفسية لدى عدد كبير من الأفراد فانه يتضح أن توزيع هذه السمة بين الأفراد إنما يتبع التوزيع الإعتدالي .

من المهم أن نضيف أنه يوجد أيضا فروق لدى كل فرد من الأفراد بين قدراته وإمكاناته واستعداداته الجسمية والنفسية والانفعالية حيث نجد أن كل فرد لديه سمة تتميز على باقي السمات الأخرى ، فمن المؤكد أنه لا يوجد فرد يمتاز في كل شيء أو فرد أخر لا يمتلك أي من المميزات. حيث تختلف الجوانب المختلفة في شخصيته ولا تكون في مستوى واحد وهذا ما يؤدى إلى أن يصبح لكل فرد أسلوبه الخاص به ، وبالنسبة للتعلم في المدرسة نجد اختلافا في قدرات التلميذ وميوله نحو تعلم المواد التعليمية المختلفة كذلك اختلاف قدرته على الاشتراك في الأنشطة والمواقف التعليمية ومدى استفادته منها، وأيضا مدى استفادته من أساليب وطرق التعليم المختلفة وهكذا. مما يفرض **دراسة سمات التلميذ المختلفة وقياسها والتعرف عليها من خلال تطبيق العديد من المقاييس والاختبارات المختلفة وذلك إذا أردنا حقا أن نحقق حاجاته الفردية ونساعده ليصل إلى أقصى تعلم يمكن أن تؤهله له هذه القدرات والسمات.**

ولما كان تحصيل التلميذ لما يقدم له من معرفة إنما يمثل وإلى يومنا هذا الهدف الأول للمؤسسات التربوية لذلك فان جميع المحاولات والدراسات التي تمت في المجال التربوي، إنما كانت للتعرف على أفضل الاستراتيجيات التي يمكن استخدامها لمساعدة كل تلميذ على التحصيل إلى أقصى درجة ممكنة عن طريق تحقيق حاجاته التعليمية الفردية.

لقد أشار بلوم (Bloom,1968) أن المعلم يبدأ تعليمه في الفصل وهو يتوقع أن يتمكن ثلث تلاميذه فقط من تعلم ما يقدم لهم بصورة مناسبة في حين أن ثلث التلاميذ الثاني سوف يخفقون في هذا التعلم ، والثلث الباقي يمكنهم أن

يتعلموا ما يقدم لهم ولكن ليس بالدرجة المطلوبة. وعادة ما ينتقل تأثير هذه التوقعات من جانب المعلم إلى التلاميذ من خلال طرق التقويم التي يستخدمها معهم، ويترتب على ذلك أن تصبح الأهداف التي يسعى المعلم إلى تحقيقها بالنسبة لكل تلميذ ثابتة لا تتغير تحدد بهذه التوقعات، وبالتالي يصبح لها تأثير مدمر على النظام التعليمي خاصة وأنها تضعف من الدافع عند التلاميذ نحو التعليم وتقلل من الإحساس بقيمة الذات.

ولكن واقع الحياة التي نعيشها بنظمها الاقتصادية وما تتطلبه من مهارات من القوى العاملة لا تسمح بالاستمرار في هذا النظام وتحتم الاهتمام بكل فرد في المجتمع واكتسابه المهارات اللازمة التي تؤهله ليصبح عضوا عاملا ومنتجا وبخاصة أن الكثير من المربين قد نادوا في كتاباتهم أن ما يزيد عن ٩٠% من التلاميذ يمكنهم أن يتقنوا ما يقدم لهم من مواد تعليمية، وبالتالي فان **وظيفة التربية والمربين تصبح التعرف على الطرق والمواد التعليمية التي تمكن أكبر نسبة من الطلاب من إتقان العلوم المقدمة لهم، ولا يتأتى ذلك بطبيعة الحال إلا إذا استطعنا أن نجد الاستراتيجية التي تساعد كل تلميذ على التوصل إلى إتقان ما يقدم له من مادة تعليمية.** وسنحاول فيما يلي التعرض بالدراسة لمفهوم **التعليم والتعلم للإتقان** :

■ أن الهدف الرئيسي من التعليم والتعلم للإتقان هو أن يصل التلميذ إلى مستوى من التحصيل لا يصل إليه عادة تحت ظروف التعليم السائدة في الفصول المدرسية التقليدية، وهو بذلك يقترب من التعليم الفردي من حيث أن هدف كل منهما هو زيادة تحصيل التلاميذ إلى أقصى درجة ممكنة تؤهلها لهم قدراتهم وذلك عن طريق **توفير شروط معينة في البيئة التعليمية التي تخطط بصورة منظمة ودقيقة بحيث تسمح بهذا التقدم في التحصيل.**

■ لكي يحقق التعليم والتعلم للإتقان الهدف المرجو منه قد يعتمد في بعض مراحل تطبيقه على توفير تعليم فردي ، وفي مراحل أخرى علي توفير تعليم تعاوني ؛ وذلك لتحقيق الحاجات الفردية للتلاميذ بحيث يمكنهم الوصول إلى مستوى الإتقان المطلوب. ولا يعني هذا أن يكون التعليم

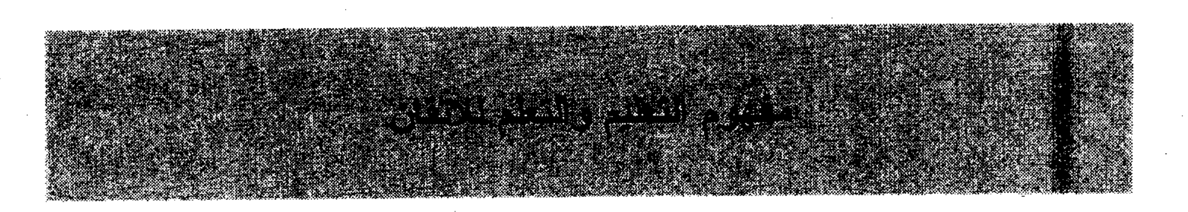
الفردي أساسا للتعلم للإتقان حيث أنه يمكن استخدام التعلم للإتقان في حالة تعلم التلاميذ في مجموعات صغيرة كانت أو كبيرة وفي هذه الحالة تستخدم وسائل أخرى لمساعدة التلميذ للوصول إلى إتقان المادة التعليمية التي يَدْرُسها ، كما أن يسمح له بالدراسة معتمداً على نفسه ودخول الامتحان عدد من المرات إلى أن يصل إلى المستوى المطلوب في التحصيل .

يعتمد التعليم والتعلم للإتقان على حقيقة هامة مؤداها : أنه إذا قدم لمجموعة من التلاميذ مادة تعليمية معينة وتعرضوا لنفس التعليم من حيث كمية التعليم ونوعيته والوقت المسموح لهم به للتعلم ، فانه من المتوقع أن يكون تحصيل هؤلاء التلاميذ موزعا في صورة التوزيع الاعتدالي المعروف حيث تكون العلاقة الإرتباطية بين تحصيل هؤلاء التلاميذ وبين قدراتهم مرتفعة نسبيا (قد تصل إلى ٠,٧) في حين أنه **إذا كان التلاميذ موزعون تبعا للتوزيع الاعتدالي بناء على قدراتهم ولكن نوعية التعليم والزمن المسموح لهم به لاستكمال عملية التعلم جعل مناسبا لصفات وقدرات كل تلميذ وحاجاته فانه من المتوقع أن يصل معظم التلاميذ إلى مستوى إتقان المادة العلمية وتصبح العلاقة الإرتباطية بين قدراتهم والتحصيل قريبة من الصفر.** وقد أكد كارول (Carroll,1963) هذه الحقيقة عندما عرف **القدرة علي التعلم بأنها الوقت الذي يحتاجه التلميذ ليصل إلى إتقان مادة تعليمية معينة، بمعنى أن العامل الرئيسي والمحدد لتحصيل التلميذ ليس قدرته العقلية ولكن ما يمكن أن توفره له من وقت يقضيه في التعلم لمادة معينة.** إذا كانت هذه الحقيقة مؤكدة فان معنى ذلك أن جميع التلاميذ يمكنهم أن يصلوا إلى المستوى المطلوب من الإتقان للمادة التعليمية. والواقع أن هناك بعض الشواهد التي تؤكد صحة هذا الفرض : منها أن أي مهارة لا يتقنها إلا التلاميذ ذوي القدرات العالية في مستوى دراسي معين من الممكن أن يصل إلى إتقانها معظم التلاميذ في المستويات الدراسية التالية، وهذا يعني أن القدرة لدى التلميذ لا تحدد قدرته على إتقان مادة تعليمية وإنما تحدد الوقت الذي يحتاجه لهذا الإتقان، ومن هنا وصل بلوم وكارول إلى أن ٩٥% من التلاميذ يمكنهم أن يتعلموا موضوعا أو مادة معينة بمستوى عال من الإتقان ؛ ويتطلب هذا بطبيعة الحال مزيدا من الجهد والوقت والمساعدة لبعض

التلاميذ؛ ويصبح التحدي الأكبر الذي يواجه المربين هو البحث والدراسة لمعرفة الطرق التي يمكن أن تؤدي إلى تقليل الوقت للمتعلم البطيء *Slow Learner* على قدر الإمكان.

وفي عام ١٩٧١ قدم بلوم نموذجا للتعليم المدرسي (Bloom, 1971) ، اعتبر فيه أن درجة تعلم التلميذ هي نتيجة مباشرة للوقت الذي يقضيه في التعلم *Time on Task* وأن الوقت الذي يقضيه كل تلميذ في التعلم يتأثر ببيئة التعلم التي بدورها تتأثر بمدخلات السلوك الادراكي أو العقلي للتلميذ *Cognitive Entry Behaviors* مثل قدراته واستعداده لهذا العمل أو المهنة التعليمية *task* وكذلك مدخلات التلميذ العاطفية *Affective Entry Characteristics* مثل اتجاهه وميله نحو العمل.

وقد قبل بلوم افتراض كارول : بأن **معظم التلاميذ يمكنهم إتقان المهارات الأساسية إذا سمح لهم بقضاء الوقت الكافي للتعلم**، وذلك إذا كان التعليم ذا نوعية جيدة بحيث يمكنهم الاستفادة منه. وقد استخدم هذا الافتراض أساسا بني عليه افتراضه بأن اختبارات الذكاء التي تقيس القدرات تشير إلى أن التلاميذ يتوزعون توزيعا اعتداليا *Normal Distribution* وأنه إذا تلقى التلاميذ نفس التعليم (أي نفس الكمية والنوعية والوقت) فان أداءهم في نهاية العملية التعليمية سوف يكون أيضا موزعا توزيعا اعتداليا.



يقصد بالتعلم للإتقان أن يصل التلاميذ إلى مستوى من التحصيل يحدد لهم مسبقا كشرط لنجاحهم في دراستهم للمقرر المقدم لهم. وعادة ما يكون هذا المستوى من التحصيل عاليا بحيث يمكن القول أنه يصل إلى مستوى الإتقان للمادة التعليمية. وعادة ما يستخدم معيار لمستوى الإتقان يسمى (معيار ٩٠/٩٠) ويقصد به توقع أن يصل ٩٠% من التلاميذ إلى تحصيل ٩٠/

من الأهداف في ٩٠/ من الموضوعات عند تقويمهم. ولكي يتمكن التلاميذ من تحقيق مستوى الإتقان المطلوب والمتوقع منهم الوصول إليه، فانه لابد من توفير شروط في بيئة التعلم تضمن وصولهم إلى هذا المستوى منها: **تقديم التعليم الجيد وتقديم المساعدة والعلاج المناسب للتلميذ في حالة وجود أية مشكلات تعوق وصوله إلى مستوى الإتقان. يرتبط بذلك أيضا توفير الوقت الكافي لكل تلميذ للوصول إلى المستوى المطلوب للإتقان وذلك بعد التأكد من وجود علاقة موجبة بين الوقت الذي يقضيه التلميذ في التعلم النشط وبين مستوى التحصيل الذي يصل إليه سواء في الدراسات العلمية أو التجارب العلمية** (Frederick & Walberg, 1980) ، (Bloom, 1979) ، (Berliner, 1979) (Rosenshine, 1979) .

نموذج التعليم والتعلم للإتقان

وقد صُمّم نموذج التعلم للإتقان ، والذي أستخدم بعد ذلك في بناء برمجيات تعليمية بحيث يزيد إلى أقصى درجة ممكنة احتمال آن ينال كل تلميذ فرصته كاملة للتعلم من تعليم فعال ويعطي الوقت الكافي لتمكينه من الوصول إلى مستوى الإتقان ؛ وقد استخدم في تصميم نموذج التعليم للإتقان مبادئ اشتقت من أعمال جودلاد (Goodlad, 1959) وأندرسون (Anderson, 1976) في المدارس الابتدائية غير المقسمة إلى صفوف دراسية *Nongraded Schools* والفصول المفتوحة. أيضا من أعمال جليسر (Glaser, 1968) في التعليم الفردي في بيتسبرج في مركز تطوير أبحاث التعلم . كذلك من الأسس التي استخدمها كل من أتكينسون (Atkinson, 1968) ، وسوبس (Suppes, 1966) في التعلم والتعلم المعزز بالحاسوب والمدار به . يضاف إلى ذلك استخدام مفهومي التقويم البنائي *Formative Evaluation* والنهائي *Summative Evaluation* الذي قدمه مايكل سكريفن (Scriven, 1967) ، ومن الأسس الهامة الأخرى التي استخدمت في وضع وتنفيذ برامج التعلم للإتقان تلك التي حددها جانيه

وبارادايز (Gangue & Paradise, 1961) ، والتي تحدد أن النجاح في إتقان تعلم وحدة من الوحدات يعتبر شرطا أساسيا *Prerequisite* لإتقان ما يليها من وحدات تتبعها في تنظيم المحتوى التعليمي. أيضا من المفاهيم المستخدمة في هذا النموذج تلك التي وضعها سكينر (Skinner, 1967) عن التعلم البرنامجي والتعزيز والتغذية الراجعة المباشرة لتصحيح المعلومات الخاطئة لدى التلميذ والتي كان لها تأثير كبير على نجاح استخدام هذا النموذج للوصول بالتلميذ إلى مستوى الإتقان.



يمر التعليم للإتقان بالمراحل التالية :

١- مرحلة تحليل وتنظيم المادة التعليمية في صورة مديولات

يقسم المحتوى التعليمي إلى مجموعة متتابعة من الموديولات طبقا للأهداف التي حددت . علي أن يشمل كل موديول من هذه الموديولات مجموعة من الحقائق والمفاهيم والمبادئ والمهارات في بناء مترابط . وذلك بهدف تسهيل تعليم وتعلم ما يقدم فيها ، وكذا معالجة جوانب الضعف لدى التلميذ قبل تراكمها مما قد يعطل قدرته على تحصيل المديولات أو الوحدات التالية . تنظم هذه الموديولات بطريقة تتابعيه هرمية ، بحيث تتيح لمن يتقن الموديول الأول منها ، أن يستخدم محتوياته في إتقان الموديول الثاني ، ومن يتقن الموديول الثاني ييسر عليه إتقان الموديول الثالث ... وهكذا ؛ مما يساعد على تأكيد التعلم ، إضافة إلى أن هذا التنظيم قد يضمن مناسبة ما يقدم من مادة تعليمية في كل موديول لمستوى التلميذ الذي أتقن أهداف الموديولات السابقة.

٢- مرحلة صياغة الأهداف السلوكية وتصنيفها

تمثل خطوة **تحديد وصياغة أهداف التعليم** خطوة أساسية في تخطيط أي برمجية تعليمية ، ونحن نستخدم عبارة أهداف التعلم ، لأن اهتماماتنا ينبغي أن تتجه إلى التعلم كناتج لعملية استخدام البرمجية في التعلم ، ولذا فإن جميع الأهداف ينبغي أن تصاغ في عبارات تبين نتائج التعلم المراد تكوينها أو تنميتها لدي المتعلمين.

ويسبق تحديد أهداف التعلم عادة خطوة اختيار نشاطات وخبرات التعلم ، ويجب أن يراعي في صياغة أهداف التعلم أن تكون قابلة للقياس ، بمعنى أن تعرفنا أنواع التعلم التي نريد أن نحققها من خلال نشاط التعليم بالبرمجية والتي يمكن في النهاية أن تحدد مدى تحقيق التلميذ لهذا التعلم.

إن تحديد الأهداف السلوكية للبرمجية التعليمية نشاط له طبيعة تطويرية، بمعنى أنه يحتاج إلى تنقيح وتعديل أو تغيير أو إضافات معينة كلما انتقلنا في دورة إنتاج البرمجية من مرحلة إلى أخري ، فقد لا تبدو أهمية الأهداف واضحة لبعض المنفذون أو المصممون إلا بعد تحديد موضوعات المحتوى وعناصره ، وفي بعض الأحيان لا تتضح أهداف التعليم الحقيقية لموضوع البرمجية إلا بعد اختيار نشاطات وخبرات التعلم المناسبة ، أو طرق وأساليب تقويم تعلم التلاميذ لهذه البرمجية ، وعلى هذا يمكن أن تبدأ بصياغة عامة للأهداف ، ثم تنتقل قدما إلى الخطوات التالية وفق تتابعها في عملية التخطيط، وبعد أن تتضح لك تلك الأهداف تماما تعود إلى كل منها مرة أخرى لكي تحددها تحديدا تفصيليا.

الأهداف هي عبارات معينة نحدد بها النتائج التي يتوقع من التلاميذ الوصول إلى تحقيقها بعد دراستهم للمحتوى ، والتي يمكن تصنيفها طبقا لتصنيف بلوم مثلا : إلي أهداف المجال الادراكي *Cognitive Domain* وأهداف المجال المهارى *Psychomotor Domain* وأهداف المجال العاطفي أو الوجداني *Affective Domain* .

٣- مرحلة بناء الاختبارات وتحديد مستوى الإتقان

يلي تحديد الأهداف بناء الاختبارات اللازمة ، وتحديد مستوى الأداء (مستوى الإتقان) الذي يُقبل من التلميذ كدليل على إتقانه للمادة التعليمية المقدمة وتحقيقه للهدف والذي يحدد نجاح التلميذ وانتقاله إلى وحدة دراسية أخرى أو إلى دراسة مادة تعليمية جديدة ؛ وعادة ما نحتاج في هذه المرحلة إلى بناء الاختبارات التالية :

١- اختبارات تسكين
٢- اختبارات تشخيصية
٣- اختبار إتقان : بنائية ونهائية

٤- مرحلة إعداد أدوات ووسائل التقدير القبلي *Reassessment*

إضافة إلى اختبارات التسكين التي تهدف إلى **تحديد النقطة التي يبدأ منها كل تلميذ تَعَلُّمه وكيفية تَعْلِيمهُ والتعامل معه** فإذا كان المنهاج مقسم إلى وحدات دراسية نسقيه قصيرة، وهو ما يحدث عادة عند استخدام التعليم والتعلم للإتقان ، فان عملية التسكين تهدف إلى تحديد أي الوحدات الدراسية التي يجب على التلميذ أن يبدأ عندها دراسته. وتهدف عملية التقدير القبلي إلى **تحديد طرق التعليم التي يجب أن تستخدم مع التلميذ في دراسته لهذا المنهاج** ، وذلك بناء على معلومات عن تحصيل التلميذ وما لديه من معلومات عن هذا المنهاج ، وما سبق له أن حققه من أهداف متصلة بالأهداف التي يتوقع أن يصل إلى تحقيقها في نهاية دراسته لهذا المنهاج ، ويستخدم في عملية التقدير القبلي هذه **نتائج التلميذ وأدائه في اختبارات التحصيل المختلفة واختبارات القدرات التي سبق أن طبقت عليه سواء عند دراسته لوحدات سابقة في نفس العام الدراسي أو في الأعوام الدراسية السابقة كذلك يستخدم فيها نتائج ملاحظات وأحكام المعلمين، وأحكامه وملاحظاته هو عن نفسه.**
هذا ويعد التقدير القبلي بالحاسوب اكثر تكاملا وشمولية حيث تتكامل اختبارات التسكين مع خصائص المتعلم ليحدد الحاسوب النقطة التي يبدأ منها كل تلميذ تعلمه وكيفية تعلمه والتعامل معه.

٥- مرحلة تحديد متطلبات التعليم القبلية *Prerequisite*

وهي المرحلة التي يتم فيها تحديد أنماط تعلم المتعلمين ، وكذا المفاهيم والمهارات اللازمة لتعلم محتوى البرمجية الحالية.

٦- مرحلة التخطيط للوصول إلى الإتقان

تبدأ مرحلة التخطيط للوصول إلى الإتقان **باختيار طرق التعليم** والأنشطة المختلفة التي تستخدم لتقديم المادة التعليمية للتلاميذ لمساعدتهم على التوصل إلى مستوى الإتقان المطلوب . فالمعيار الأساسي هنا لاختيار أي من هذه الطرق هو مدى جدواها في وصول المتعلمين إلى مستوى الأداء المطلوب ، وفي كثير من البرمجيات التعليمية التي استخدم فيها مدخل التعليم والتعلم للإتقان كانت الأهداف ونتائج التعلم تحدد في صورة مستويات مختلفة للأداء يتوقع أن يصل إليها التلاميذ، وفي كل مجال من مجالات الأداء يحدد حدا أدنى للنجاح *Minimum Pass Level* وهذا الحد الأدنى يقرر أقل مستوى يجب أن يصل إليه التلميذ ليتمكن من أداء السلوك المقبول في هذا المحتوى وليتمكن من النجاح والانتقال إلى محتوى تعليمي آخر أو إلى وحدة تدريسية أخرى من نفس المحتوى.

ونظرا لأن التلاميذ يتعلمون بسرعات مختلفة ، كما وأن أي من طرق التعليم المستخدمة قد لا تكون نافعة لكل تلميذ بنفس الدرجة ؛ لذلك فان تعليم مختلف لكل تلميذ لابد وأن يتيح لكلٍ منهم أن يصل إلى النتائج المرغوبة الخاصة به ، وإذا كان من المهم أن يكون التعليم مناسبا ومؤثرا فان الوقت الذي يحدد للعملية التعليمية يجب أن يحدد تبعا لحاجات كل تلميذ ؛ بمعنى أنه يجب أن يسمح للتلميذ أن يستمر في تلقي التعليم حتى يصل إلى تحقيق الأهداف الخاصة به ، كما أنه لابد من توفير بدائل من طرق التعليم وأنشطة التعليم لاستخدامها عند الحاجة.

وان تقويم تحصيل التلميذ للحد الأدنى من الأداء يُعَدُّ ذو أهمية كبيرة في البرمجيات التعليمية أولا لكي يمكن اتخاذ القرار الخاص بانتقال التلميذ إلى دراسة وحدة دراسية أو محتوى آخر ، وثانيا للتعرف على التلاميذ الذين لم يصلوا إلى واحد أو أكثر من مستويات الإتقان المحددة بالبرمجية ، وبالتالي يمكن وصف التعليم المناسب لهم ، الذي يساعدهم في الوصول إلى مستويات الإتقان هذه . وفي هذه الحالة – حالة استخدام التعليم والتعلم للإتقان – يجب أن يكون التركيز على تعلم المهارات والحقائق والمفاهيم والاتجاهات الأساسية ، إضافة إلى إتاحة الفرصة للتلميذ لكي يتعلم عن طريق **الاستكشاف والاستقصاء** ولابد من توفير التوازن بين هذين الجزئيين من المنهاج أي الجزء الذي يجب أن يكتسبه التلميذ بإتقان والذي يمكن تسميته بقلب أو محور المنهج *The Core of the Curriculum* والجزء الثاني الذي يترك فيه الفرد ليكتشف بنفسه ويتعلم عن طريق الاستقصاء.

إن القدرة على تقبل التعليم إنما تعرف بأنها قدرة التلميذ على فهم طبيعة المهمة التعليمية التي تقدم له والطرق التي يتبعها في تعلم هذه المهمة. والواقع أنه لا توجد طريقة واحدة للتعليم يمكن أن يقال إنها الطريقة التي تؤدي بكل متعلم إلى الوصول إلى مستوى الإتقان ولكن إذا توفرت طريقتان أو أكثر تصلح لمساعدة التلميذ للوصول إلى هذا المستوى فان على البرمجيات التعليمية الذكية أن تختار الطريقة التي تتفق مع رغبة التلميذ وتفضيله ؛ طبقا لما يتوافر بها من معلومات من خلال التقدير القبلي.

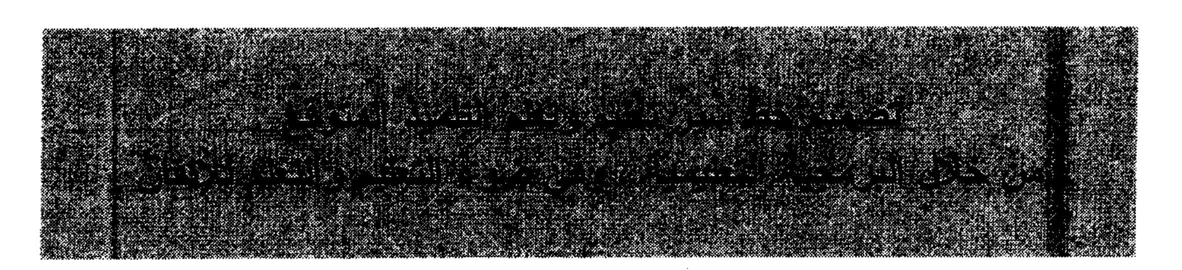
وإذا كان المجال التربوي مازال يحتاج إلى إجراء العديد من الأبحاث والدراسات حول كيفية ربط الفروق الفردية بين المتعلمين بالاختلافات في نوعية التعليم إلا أن هناك بعض الأدلة تشير إلى أن **بعض التلاميذ يمكنهم التعلم بصورة جيدة من الطرق التي توفر لهم الاعتماد على النفس، في حين أن البعض الآخر يتعلم من الطرق والمواقف التعليمية التي يتم تنظيمها بدرجة عالية. كما أن بعض التلاميذ يتعلمون من التوضيحات الحسية أكثر من غيرهم، كذلك قد يحتاج البعض إلى استخدام العديد من الأمثلة أثناء**

التعليم حتى يمكنهم التعلم، كما قد يحتاج البعض إلى المزيد من التحفيز والدافعية الخارجية ؛ وهذا كله يعني أن طريقة التعليم التي قد تكون مناسبة لنوعية من التلاميذ قد لا تكون مناسبة لنوعية أخرى منهم. وتتطلب مرحلة التخطيط للوصول إلى الإتقان الخطوات التالية :

- اختيار طرق التعليم وإعداد الأنشطة والمواد التعليمية .
- إعداد خطة عرض المحتوى الوقائية.
- إعداد خطة العرض العلاجية.
- إعداد خطة العرض الإثرائية.

إن كل مرحلة من مراحل التعليم والتعلم للإتقان السابقة لها أهمية خاصة ودور في مساعدة التلاميذ على إتقان المهارات والحقائق والمفاهيم الأساسية للمنهاج ، كما وأن كل مرحلة أيضا تلعب دورا هاما في تنظيم المحتوى بكفاءة ؛ بما يضمن أن يبدأ التلاميذ دراستهم عند المستوى المناسب لكل منهم ، ويتقدمون فيها كلٍ حسب سرعته الخاصة وتعريض كلٍ منهم إلى المواقف التعليمية التي تضمن بأكبر درجة ممكنة وصولهم إلى تحقيق الأهداف.

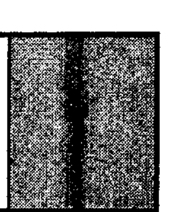
ويعتقد البعض أن التعليم والتعلم للإتقان يمكن أن يؤدي إلى تخريج تلاميذ متماثلين لديهم نفس المعرفة ونفس المهارات ويستخدمون نفس طرق التفكير ، ولكن الحقيقة أن هذا التعليم ربما كان ناتجة تلاميذ يختلفون فيما بينهم عن أولئك الذين تعلموا بالطريقة التقليدية التي تستخدم للوصول إلى هذه الأهداف أو النتائج ، وإذا كان هناك تشابه أو تماثل بين النتائج التي يصل إلى تعلمها التلاميذ فان هذا التشابه أو التماثل يتمثل في الحد الأدنى من المعرفة الأساسية ثم وبالإضافة إلى هذا المستوى الأدنى من المهارات الأساسية فان كل تلميذ يستطيع أن يتقدم إلى تحقيق الأهداف التي تناسب حاجاته وميوله الخاصة به.



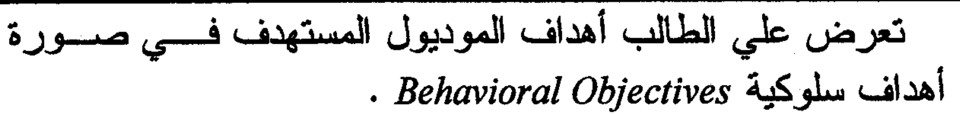
في الحالة المثالية

ينبغي علي المصمم أن يتوقع خط سير تعليم وتعلم التلميذ من خلال البرمجية التعليمية ، أمام شاشة الحاسوب مع وضع ماهية ومفهوم التعليم والتعلم للإتقان في الاعتبار ، حيث ينصب الاهتمام على حسن إدارة عملية التعلم *Instruction Management* وليس على إدارة التلميذ *Student Management* ، وهنا ينبغي علي المصمم والمخطط للبرمجية أن يبذل قصارى جهده في كيفية تحفيز التلميذ للتعلم وإمداده بالمواد التعليمية المتنوعة ، وان يحدد كيفية تحديد مدى تقدم كل تلميذ في تعلمه وتشخيص صعوبات التعلم لديه وتوفير العلاج المناسب له ، **ويتم ذلك طبقا للترتيب التالي :**

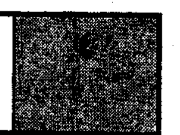
يبدأ الطالب العمل بالبرمجية بتشغيلها وتسجيل اسمه وكافة المعلومات التي تُطلب منه ومن المشرف علي تعليمه من المعلمين، ليبدأ الحاسوب عن طريق البرمجية بتحليل هذه المعلومات وتحديد قرارات وأفعال التقدير القبلي *Reassessment* .



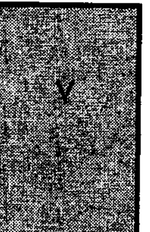
تعرض علي الطالب قوائم المحتوى التعليمي المستهدف في صورة وحدات *Units* وموديولات *Modules* ، ليختار التلميذ الموديول الذي يرغب في دراسته .



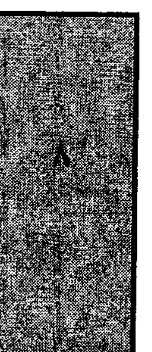
تعرض علي الطالب أهداف الموديول المستهدف في صورة أهداف سلوكية *Behavioral Objectives* .

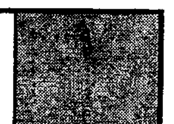

يقدم للطالب اختبار التسكين *Placement Test* الخاص بالموديول الذي سبق له وأن اختاره .


طبقا لنتيجة اختبار التسكين يسكن التلميذ عند نقطة بداية تعلمه بالموديول المستهدف .


يبدأ الطالب تعلمه بالموديول المستهدف ، من النقطة التي سبق تحديدها – طبقا لنتيجة اختبار التسكين – وذلك بتلقيه محتوى المتطلبات السابقة واللازم للسير قدما في تعلمه للموديول المستهدف.

يتلقى الطالب المحتوى المستهدف للموديول ، من خلال العروض التفاعلية ذات الوسائط المتعددة والشروح المحفزة لدافعيته ، والأمثلة المحلوله والتدريبات مع توافر التغذية الراجعة الفورية الموجبة والسالبة ؛ طبقا لما تم إعداده بالخطة الوقائية.

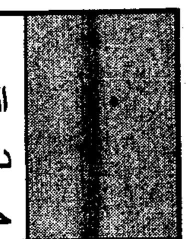

يُقدم للطالب الاختبار التشخيصي *Diagnostic Test* الخاص بالموديول الذي سبق له وأن اختاره ، وانتهي من محاولة دراسته ، وذلك بهدف جمع معلومات مفصلة عما تعلمه الطالب : ما حققه من أهداف وما أخفق في تحقيقه، والوصول إلى توصيف دقيق لحالة الطالب *Student Profile* لتوضيح ما يعاينه الطالب من ضعف وما حققه من أهداف .


بناء علي نتيجة الاختبار التشخيصي للموديول ، يتم توجيه الطالب إلى الأنشطة المناسبة لحالته : علاجية أو إثرائية.

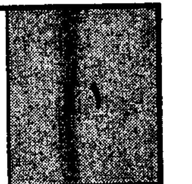
➢ فإذ لم يحقق الطالب أهداف الموديول المستهدف ؛ يقدم لـــه المحتـــوى التعليمي الذي اخفق في تحقيق أهدافه في صورة عروض تفاعلية ذات وسائط متعددة وشروحات مختلفة عما سبق تقديمه بالخطة الوقائية ؛ مع أمثله محلولة وتدريبات وتوافر تغذية راجعة اكثر استثارة لدافعيته ، مناســـبة لخصائصـــه والتي تم الحصول علي معلومات وفيرة عنها عن طريق وسائل التقدير القبلي – وتشبه هذه الخطوة وصفة العلاج التي يضعها الطبيب ولذلك فقـــد أطلـــق عليها *Prescription* – ؛ وذلك طبقا لما تم إعداده **بالخطة العلاجية** .

وفي حالة ما إذا ثبت أن الطالب لا يملك القدرات والمهارات التي تُمكنـــه من دراسة هذا المستوى بنجاح أو عندما يصبح واضحا أن اختيار هدف جديد آخر له يكون أنسب من الهدف الذي تم اختياره له مسبقا. عندئذ يعاد وضـــع الطالب عند نقطة بداية أخرى وهو ما أطلق عليه بعمليـــة إعـــادة التسـكين *Relocation* حيث يبدأ الطالب في دراسة موضوع آخر، أو توفر له محتـــوى تعليمي خاص لاستكمال الخبرات والمهارات التي تنقصه قبل أن يعود إلـــى استكمال دراسته مرة أخرى.

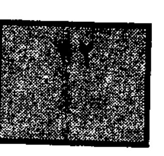
➢ أما إذا ما حقق الطالب أهداف الموديول المستهدف ؛ يقدم له أنشـــطة ومواد تعليمية بغرض إثراء عملية التعليم *Enrichment Materials* ؛ وذلـــك طبقا لما تم إعداده **بالخطة الإثرائية** .



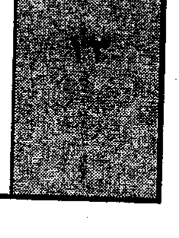
يقدم للطالب اختبار الإتقان البنـــائي *Formative Mastery Test* الخاص بالموديول الذي سبق له وأن اختاره ، وانتهي من محاولـــة دراسته ، وتلقى فيه مواد تعليمية وأنشطة علاجية أو إثرائية حسب حالته .



يزود الطالب أو المشرف على تعليمه من المعلمين أو ولي أمره بتقرير مفصلا عن درجة أدائه ومدى إتقانه للأهـــداف ، لاتخـــاذ القرار المناسب حسب حالة الطالب.



ينتقل الطالب – وبنفس الطريقة – إلى تعلم الموديول الثاني ثم الثالث ... وهكذا ؛ حتى ينتهي من تعلم كافة موديولات الوحدة.



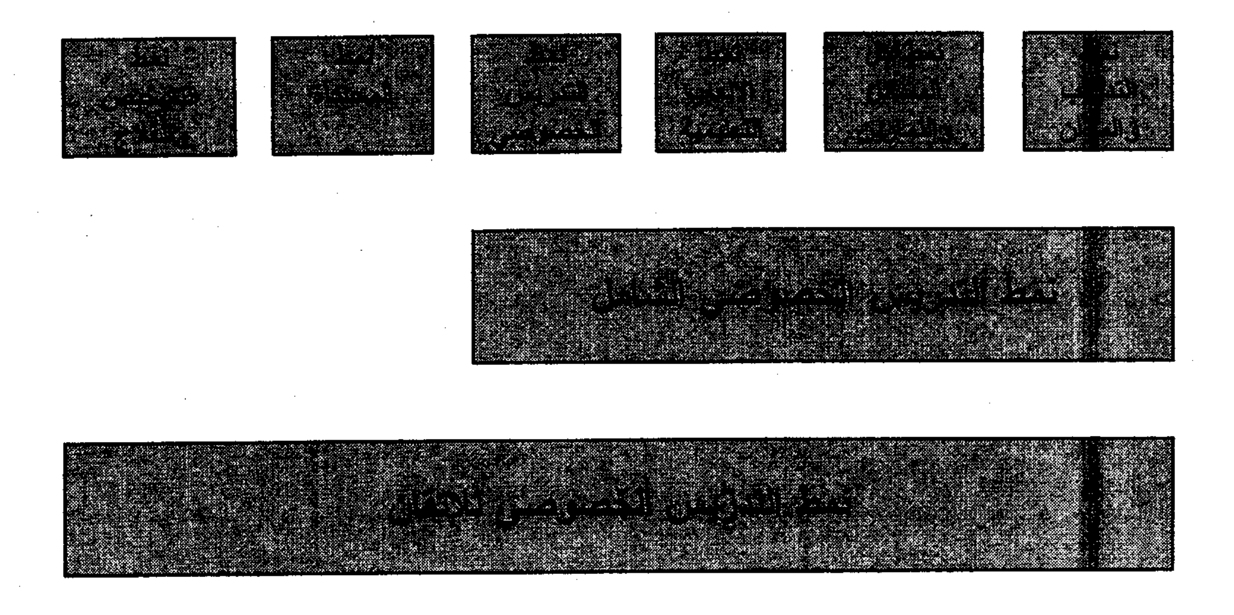
يتلقى الطالب اختبار الإتقان النهائي *Summative Mastery Test* للوحدة ، حيث يزود الطالب أو المشرف علي تعليمه من المعلمين أو ولي أمره بتقرير مفصلا عن درجة أدائه في الوحدة ، ومدى إتقانه لأهدافها لاتخاذ القرار المناسب حسب حالته.

١٤

ينتقل الطالب – بنفس الطريقة – إلى تعلم الوحدة الثانية ثم الثالثة ... وهكذا ، حتى ينتهي من تعلم كافة الوحدات المستهدفة .

هذا وقد يحتاج مصمم البرمجية قبل البدء في عملية التصميم والتخطيط للبرمجية أن يتعرف علي أنماط البرمجيات التعليمية ومراحل تطورها.

يمكن تحديد أنماط البرمجيات التعليمية *Educational Software* حسب هدف كل من التلميذ والمعلم من استخدامها ؛ فقد صنفها كل من فيتر (Fetter,1984) ، ووالاس وروز (Wallace & Rose, 1984) حسب درجة تفاعل التلميذ معها ؛ في حين صنفها كل من هولدن (Holden, 1987) وديكي (Dickey, 1988) حسب كمية مساعدتها للمعلم في تكملة أدواره : **فهناك من البرمجيات التي تقدم مساعدات محدودة للمعلم ، وهناك من تقدم مساعدات معقولة وهناك من تقدم مساعدات كبيرة للتلميذ لدرجة أنه يعتبرها عوضا عن المعلم في بعض الأحيان** ، والشكل رقم (٢) يوضح أنماط برمجيات التعليم والتعلم والتدريب وتطورها.

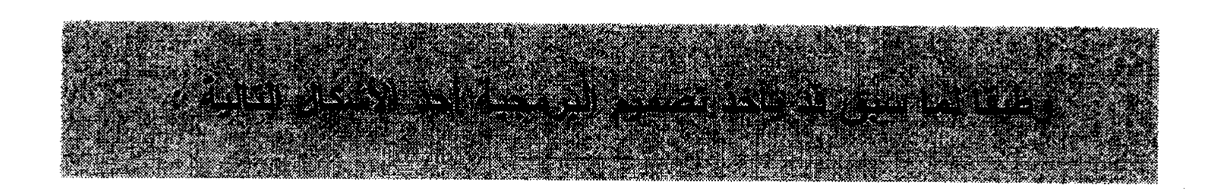


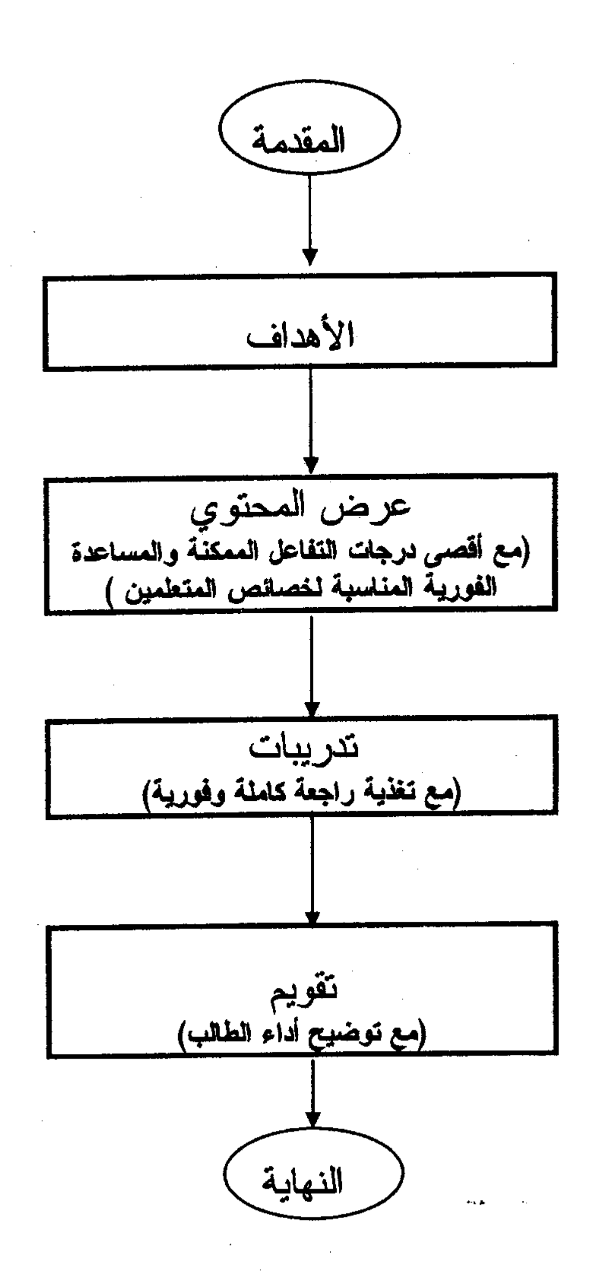
شكل رقم (٢)
تطور أنماط برمجيات التعليم والتعلم والتدريب

واضح من الشكل السابق أن البرمجيات التعليمية : للتعليم والتعلم والتدريب بدأت بنمط التدريب والمران *Drill & Practice* الأكثر تواضعا ، ثم ظهر نمط حل المسائل والتمارين *Problem Solving & Exercise* ، ثم نمط الألعاب التعليمية *Instruction Games* ، ثم نمط التدريس الخصوصي *Tutorial* ثم نمط المحاكاة وتمثيل المواقف *Simulation* ، ثم نمط التشخيص والعلاج *Diagnostic / Proscriptive* ؛ كل في صورة برمجيات مستقلة تماما عن الأخرى لقصور أجهزة الحاسوب وطاقات تخزينه لمتطلبات دمج تلك البرمجيات معا .

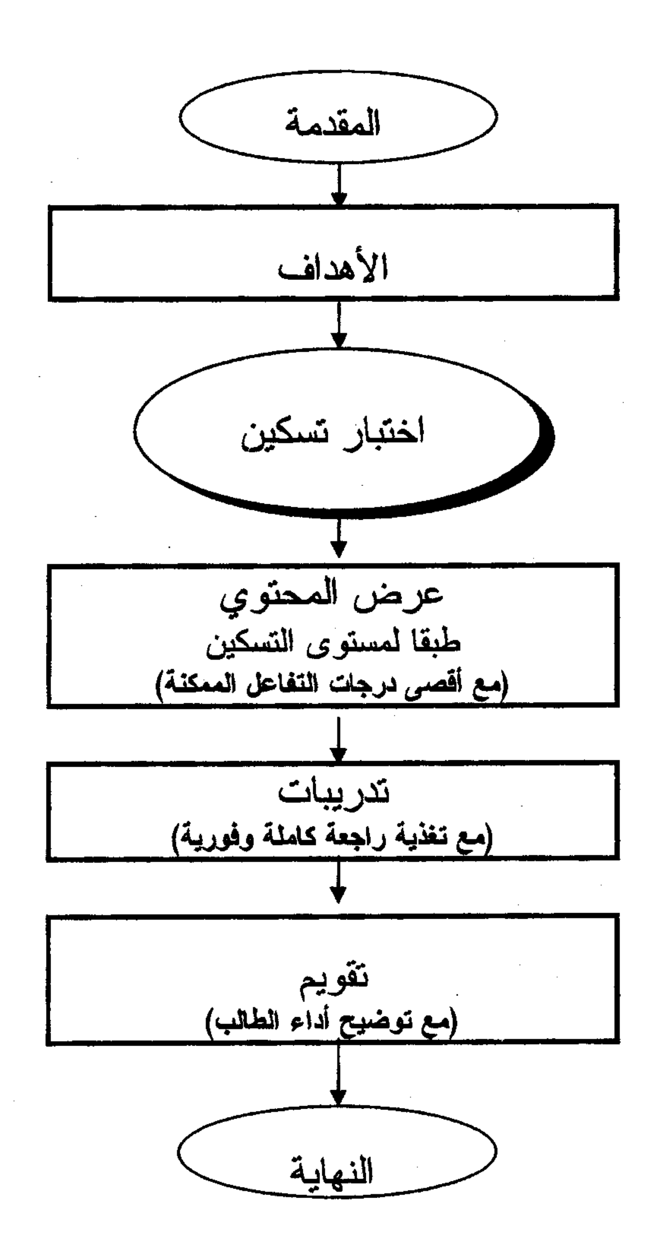
مع تطور تكنولوجيا الحاسوب ، وتعاظم طاقاته التخزينيه ، وزيادة سرعته في تناول المعلومات ، ظهرت برمجيات معقدة عرفت بنمط **التدريس الخصوصي الشامل** *Full Tutorial Instruction* التي دمجت أنماط : التدريب والمران *Drill & Practice* وحل المسائل والتمارين *Problem Solving & Exercise* ، والألعاب التعليمية *Instruction Games* إضافة إلى نمط التدريس الخصوصي *Tutorial* **كبرمجية واحدة تتميز بالتفاعل والفاعلية.**

ثم سرعان ما حدث التطور الأعظم بظهور أنماط جديدة من البرمجيات غاية في التعقيد والنضج تعرف **بنمط التدريس الخصوصي للإتقان** ***Full Tutorial for Mastery*** والذي عرف **بنمط إدارة العملية التعليمية بالكامل** *Computer Management Instruction* (CMI) ، والتي تطورت بدورها الآن إلي ما يسمى بنظام التعليم والتعلم المتصف بالذكاء المدار بالحاسوب *Intelligent Computer management Instruction* (ICMI)



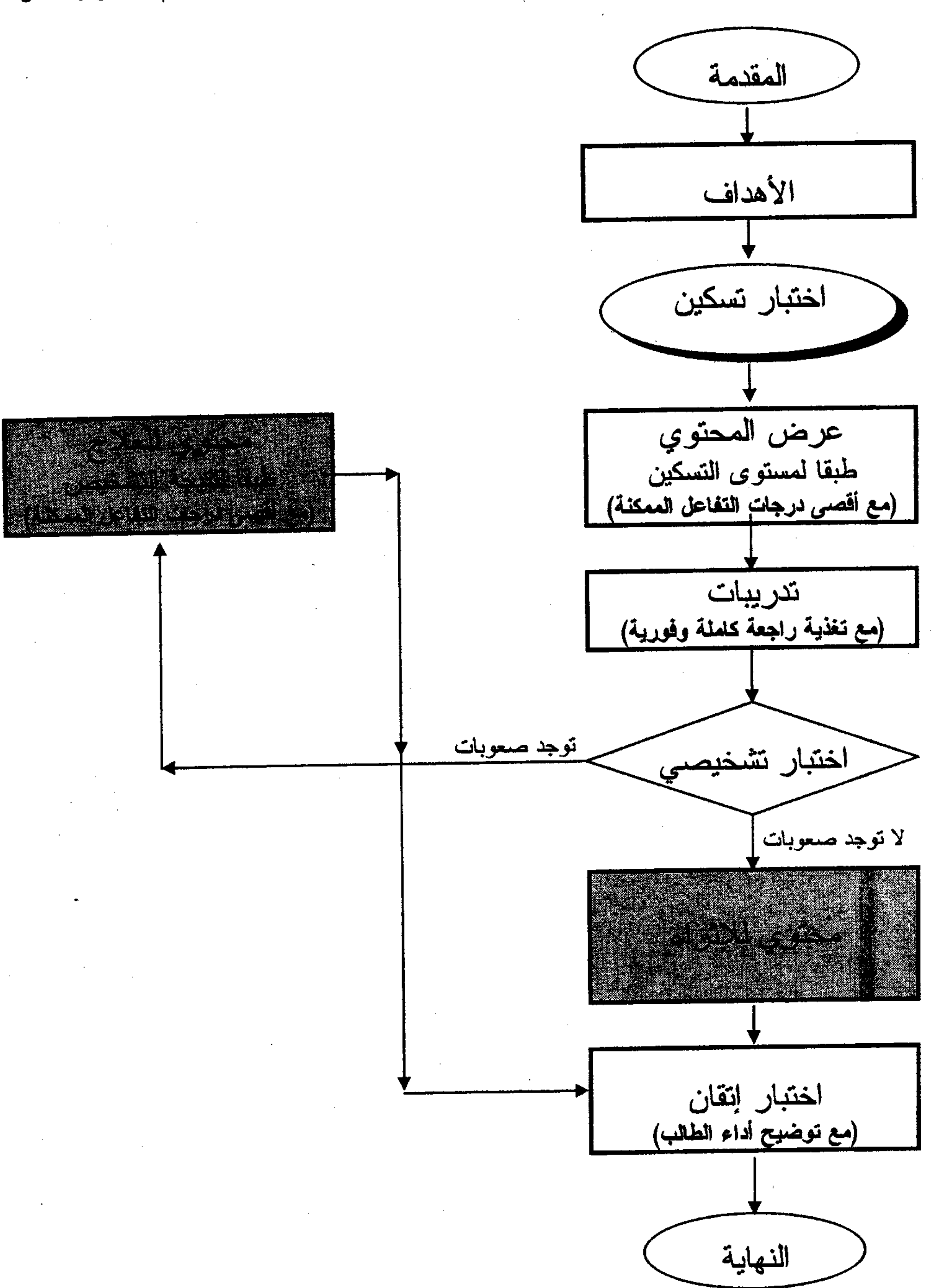


شكل (٣)
يوضح ابسط مستويات تصميم دروس البرمجية

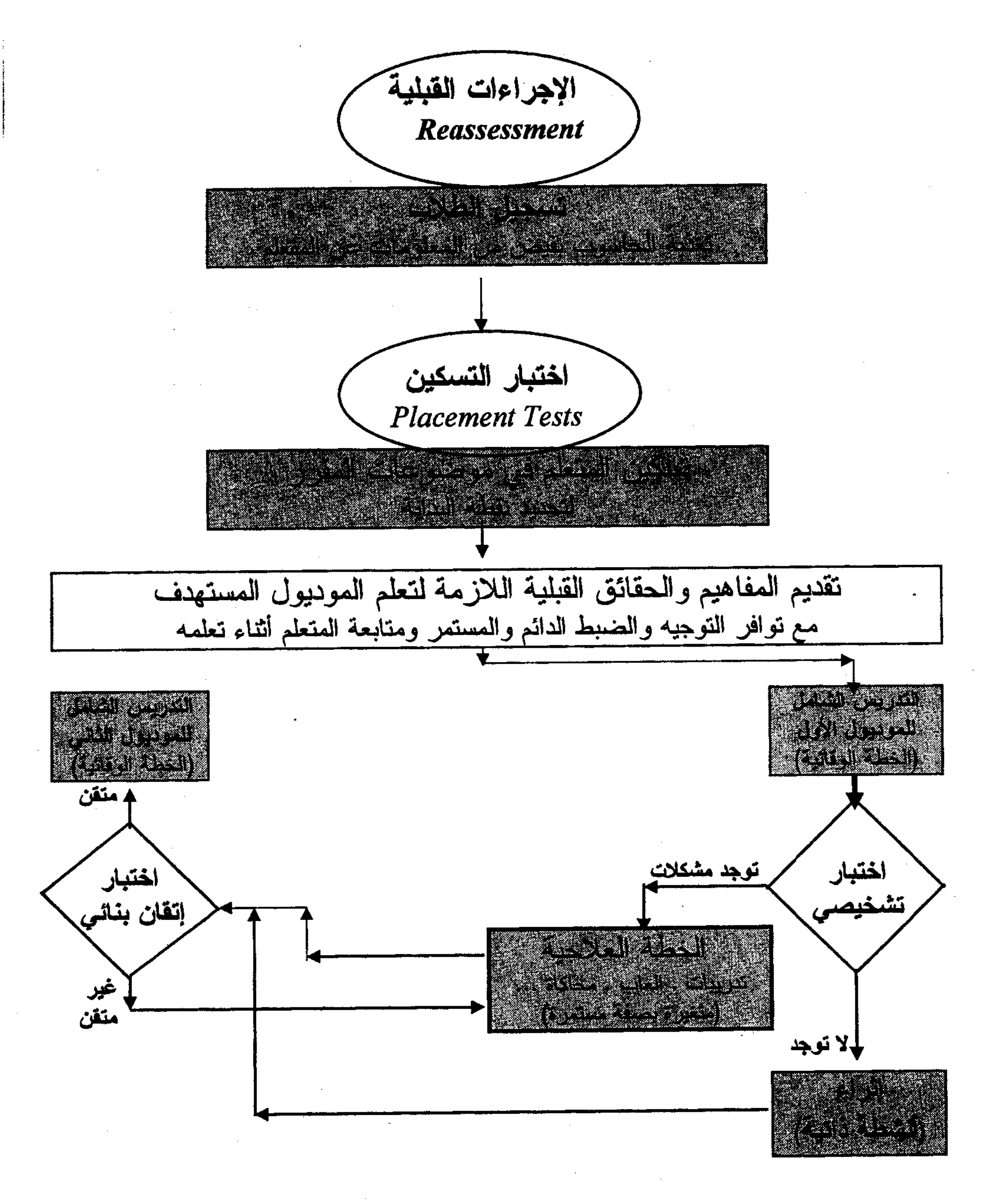


شكل (٤)

يوضح مستوى آخر لتصميم دروس البرمجية يحتوي علي اختبار تسكين



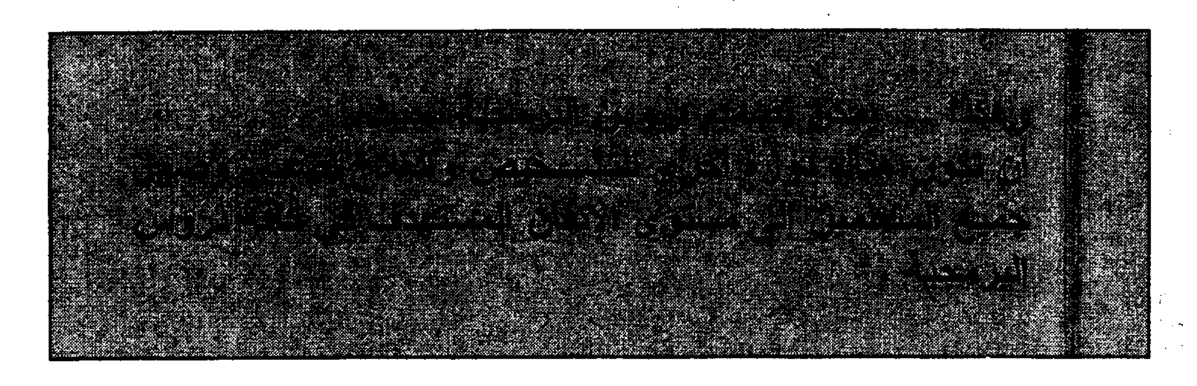
شكل (٥)
يوضح مستوى ثالث لتصميم دروس البرمجية



شكل (٦)

يوضح مستوى مثالي لتصميم تعليم أحد مديولات البرمجية

وهو الذي يحتوي علي اختبار تسكين واختبار تشخيصي وعرض جديد للمحتوي ومعلومات إثرائية وكذا اختبار إتقان (دورة واحدة للتشخيص والعلاج)



ومما هو جدير بالذكر ، أن دور مصمم البرمجية لا ينتهي عند وضع هذا التصور؛ فعليه أن يتابع تنفيذ العمل ، وقد يقوم نتيجة لهذه المتابعة بإجراء بعض التعديلات على مقترحاته السابقة ، كما أنه يقوم بالرد على استفسارات القائمين بالعمل بالمراحل الأخرى ، ويقوم بتوعيتهم ، ويوضح الأساس العلمي الذي وضع عليه تصوره السابق ، وفى بعض الأحيان يمتد دوره الذي تمثل في وضع التصور والتصميم الأساسي للمنهج ومكوناته إلى ما بعد الانتهاء من تنفيذ البرمجية . فقد يقوم بملاحظة أداء الطلاب وتفاعلهم في الموقف التعليمي الفعلي في مرحلة التجريب والتطوير ، وقد يقوم باقتراح مزيدا من التعديلات والتنقيحات التي يرى أنها ضرورية في ضوء مشاهداته وملاحظاته والمعلومات التي قام بجمعها من المعلمين القائمين على تنفيذ ذلك المقرر.

هذا وتعتبر مرحلة الإعداد والتجهيز لإنتاج البرمجية فى غاية الأهمية حيث من خلالها يتكون لدى المبرمج تصور كامل عن شكل ونمط البرمجية قبل الشروع في برمجتها وتحويلها الى منتج ، هذا وتمر هذه المرحلة بست خطوات اساسية يمكن تلخيصها كالتالي :

١- التخطيط للمنهاج

- صياغة الأهداف.
- تحديد استراتيجية التعليم والعرض .
- تسلسل محتوى المقرر.
- تحديد خصائص المتعلم.

٢- توفير مصادر المادة التعليمية

- تجميع الكتب والمراجع ذات العلاقة بالمحتوى.
- توفير الكتاب المدرسيى ودليل المعلم.
- الإطلاع على برمجيات تعليمية مشابهة.
- الإطلاع على طرق التقويم المتبعة لتقويم أداء المتعلم بالمحتوى موضوع البرمجية.

٣- تحديد الدرس

- تحليل محتوى المقرر بهدف تحديد المفاهيم والمصطلحات والأساسيات والمهارات التى يشتمل عليها المحتوى.
- تقسيم المحتوى الى موضوعات والموضوع الى دروس والدرس الى فقرات .
- تحديد نوع كل فقرة
- تحديد التفرعات برسم مخطط مسار للدرس.

٤- اختيار انواع فقرات الدرس

- تحديد قائمة بنود الدرس.
- تحديد فقرات النصوص المعنية بالعرض.
- تحديد فقرات التمارين ومفردات الأسئلة وانواعها:

- اختيار من متعدد.
- سؤال وجواب .
- صواب أم خطأ.
- مطابقة قائمتين.
- ترتيب قائمة.

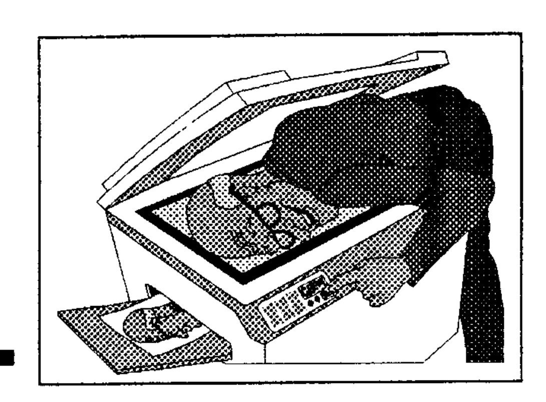
٥- تصميم فقرات الدرس

- تحديد النصوص والأشكال.
- إضافة عنصر التفاعل الإيجابي .
- إضافة المؤثرات بهدف جذب إنتباه المتعلم كالألوان والصور التوضيحية والحركة والمؤثرات الصوتية.
- تحديد العلاقة بين الفقرة وماقبلها ومابعدها.

٦- إعداد الدرس

- ملأ النماذج الخاصة بالشاشات بهدف تحديد موقع الأغراض وحجمها وخطوطها والوانها بالنسبة لوعاء كل شاشة.





الفصل الثاني

مرحلة

كتابة سيناريو البرمجيه



تمهيد

يحتاج كتابة السيناريو *Scenario* إلى بعض المهارات ، ولا نتوقع أن تصل إلى مستوى أداء متقن دون مرحلة تجريب جادة خلال سلسلة من المحاولات والأخطاء ، ولكن بمجرد أن تستوعب المبادئ الأساسية ستنطلق في بلورتها واستحداث أساليب خاصة بك . إن عرض المادة التعليمية على شاشة الحاسوب يتطلب منك الالتزام بأنماط متسقة وإحساس دقيق بالصورة التي ستبدو بها المادة التعليمية على شاشة الحاسوب ؛ وهو ما يسمى **بسيناريو الدرس**. وبالطبع لا يمكنك اكتساب هذه الخبرة إلا بالعمل على الحاسوب حيث يفوق التجريب العملي مراحل التصور النظري مهما كان دقيقا وشاملا ، ولكن قبل أن تذهب إلى الحاسوب ضع أفكارك وبلورها على الورق حتى لا يضيع الوقت أمام الحاسوب في دوامة التعديل وإعادة التنظيم .

إن السيناريو هو المزيج من شمولية الفكرة ومراعاة التفاصيل الدقيقة لتنفيذها ونقلها لعالم الواقع ، وان السيناريو المشوش سيؤدى حتما إلى برمجية رديئة . فالمقصود بمرحلة كتابة سيناريو البرمجية ، هي المرحلة التي يتم فيها ترجمة الخطوط العريضة التي وضعها مصمم البرمجية إلى إجراءات تفصيلية مسجلة على الورق . وعادة ما يقوم بمهمة كتابة سيناريو البرمجية أفضل المعلمين خبرة في المادة العلمية وطرق تدريسها ، وأن يكون معلما قديرا مشهودا له بالكفاءة علما وتدريسا وأن يكون على دراية بالمداخل المختلفة لتدريس كل موضوعات هذه المادة ، والذي يكون علي دراية أكثر من المصمم فيما يتعلق بإمكانات الحاسوب ولغة التأليف المقرر استخدامها ، وأن يكون علي اتصال دائم بالمصمم . ويمكنه الاستعانة في هذه الحالة ببعض المراجع والكتب التي تتناول تدريس الموضوعات التي تشتمل عليها المادة ، وكذا كافة ما تم تجهيزه بمرحلة الإعداد .

ويتلخص العمل في هذه المرحلة بتسجيل ما ينبغي أن يعرض على الشاشة على نماذج خاصة تعرف بنماذج السيناريو ، وهي مصممة ومقسمة بطريقة تشبه تماما شاشة الحاسوب ، وهي تختلف حسب نوع شاشة

البرمجية. إن القائم بكتابة سيناريو البرمجية قد يقوم بتحديد المواقع على الشاشة التي ستكتب فيها معلومات معينة ، مسترشدا في ذلك بأبعاد الشاشة ومساحتها، وبالإضافة إلى ذلك فإنه يحدد تسلسل ظهور هذه المعلومات والفواصل الزمنية بين كل معلومة وأخرى ، كما يحدد المعلومات التي ينبغي أن تظل على الشاشة لفترة معينة والمعلومات التي ينبغي اختفائها في أوقات محددة .

وبالإضافة إلى ما سبق فإنه يحدد نوع التغذية الراجعة *Feedback* التي ينبغي توفيرها بعد استجابة التلميذ للأسئلة التي تعرض عليه ؛ كأن يحدد على الورق ما يلي:

- عند الإجابة الصحيحة تعرض كلمات تشجيعية مثل : ممتاز، رائع، جيد جدا ، بارك الله فيك ... الخ ، وقد تكون مسموعة أو يظهر علي الشاشة منظر معبر حي أو كرتون أو يسمع صوت معبر علي السرور أو يشاهد لقطة فيديو معبرة.

- عند الخطأ لأول مرة (خ١) تعطى التغذية الراجعة ذات المستوى الأول بالشكل الموضح بالشكل السابق ، حيث يطلب من التلميذ التفكير في مسألة القسمة على اعتبار أنها مسألة ضرب مثلا.

- عند الخطأ للمرة الثانية (خ٢) تعطى التغذية الراجعة ذات المستوى الثاني .

ويستطيع معد سيناريو البرمجية أن يضع تفاصيل أكثر على الورق : مثل اختيار الألوان وما يجب تلوينه ، واختيار الأصوات والنغمات الموسيقية. إذا كان ذلك ضروريا بالنسبة للبرمجية ، مع تحديد توقيت إصدار الأصوات أو النغمات الموسيقية.

ومن الأعمال الأساسية التي يحددها معد سيناريو البرمجية تحديد عدد الأمثلة، والأسئلة في التدريبات ونوع المعلومات التي ينبغي توفيرها عقب الانتهاء من التدريب مثل عدد الأسئلة التي أعطيت وعدد الإجابات الصحيحة والوقت المستغرق أحيانا .

وعادة ما يشارك – في مثل هذه القرارات الهامة – مصمم البرمجية وقد تكون محددة أصلا في التصميم . وفى أغلب الأحيان يعود معد سيناريو البرمجية إلى المصمم عدة مرات للاستفسار عن أمور قد تكون غامضة أو أمور لم يشر إليها في التصميم ، أو أمور تكشفت عند الدخول في عملية كتابة السيناريو ، ومثل هذا التفاعل المشترك يعد من الأمور العادية المتوقعة التي نشجع على حدوثها باستمرار.

وبالنسبة للاختبارات ؛ فإن معد سيناريو البرمجية يقوم أيضا بتحديد نوع الأسئلة وعددها وكتابة مفرداتها ويحدد موقع عرض السؤال على الشاشة والبيانات المصاحبة ؛ مثل رقم السؤال وعدد الإجابات الصحيحة في بعض الأحيان ، والوقت المستغرق ومعايير الاختبار مثل الحد الأقصى المسموح به من الزمن والحد الأدنى لعدد الإجابات الصحيحة . ويسبق ذلك بالطبع كتابة تعليمات الاختبار التي قد تعطى قبل البدء الفعلي للاختبار ، وبعضها الآخر قد يصاحب عرض الأسئلة ؛ مثل وظائف بعض المفاتيح الخاصة أو طريقة إدخال أرقام عشرية أو كسور اعتيادية ، وكيفية مسح الإجابة إذا أدرك التلميذ أنها خطأ قبل الضغط على المفتاح أو الموقع المحدد مسبقا للانتقال إلى الشاشة التالية أو كيفية الرسم بالضغط على مفاتيح معينة .

بالإضافة إلى ما سبق ؛ فإن معد السيناريو يحدد طرق وأساليب جمع البيانات الخاصة بالأداء حسب التصميم الموضوع . وقد يكون التصميم قائما على عرض نتيجة الاختبار على التلميذ بمجرد الانتهاء منه ، ويطلب من التلميذ، مثلا، تسجيل هذه النتيجة على الورق تحت إشراف المعلم ، أو قد يكون التصميم قائما على أساس عرض النتيجة على التلميذ في نهاية الاختبار وتبليغه بما ينبغي القيام به في ضوء هذه النتيجة ، وتسجل النتيجة على القرص المشتمل على البرمجية التعليمية نفسه، أو على قرص آخر خاص بتجميع البيانات ، ويكون بشكل أوتوماتيكي ، ويستطيع المعلم أن يحصل على نسخة مطبوعة أو مرئية على الشاشة خاصة بنتيجة تلميذ معين أو أكثر في أي وقت يشاء . هذا وعلي معد سيناريو البرمجية القيام بما يلي :

- تحديد النصوص والأشكال ومواقعها علي الشاشة.
- تحديد عناصر التفاعل .

- تحديد المؤثرات بهدف جذب انتباه المتعلم كالألوان والصور التوضيحية والحركة والمؤثرات الصوتية.
- تحديد العلاقة بين الفقرة وما قبلها وما بعدها.
- تحديد كيفية الانتقال من شاشة إلى أخري.
- تحديد عدد الشاشات وتسلسلها .
- تحديد سلوك المتعلم المتوقع عند التعامل مع كل شاشة .

وينبغي أن **لا نبخل بالوقت الذي يستغرق في مرحلة كتابة السيناريو ، فكلما كان السيناريو معدا بطريقة واضحة ومفصلة آخذا في الاعتبار جميع الاحتمالات الممكنة ؛ كان الوقت المستغرق في عملية تنفيذ البرمجية أقل بكثير.**

قواعد عامة
ينبغي مراعاتها عند كتابة سيناريو الدرس

فيما يلي سنورد قائمة ببعض القواعد والأساليب والإرشادات العامة التي ينبغي مراعاتها عند سيناريو الدرس :

- ✓ تخلص من أسر المادة المطبوعة فالبرمجية ليست ترجمة مباشرة لمضمون الكتاب.
- ✓ خصص لكل فقرة موضوعا أو فكرة واحدة.
- ✓ ركز على النقاط الرئيسية والصعبة ، لا تستسلم لعادة زيادة الأبسط توضيحا وإغفال الأعقد لصعوبة تناوله أو عرضة.
- ✓ عمق العرض والشرح من خلال الأمثلة الكافية ، والتكرار المحسوب وإعادة طرح الأفكار بصيغ مختلفة .

- ✓ حافظ على مقروئية شاشتك ، لاتزحمها بالنصوص والأشكال ، اترك هوامش وفواصل كافية.
- ✓ أبرز الأجزاء الهامة من النصوص والأشكال باستخدام الألوان وتغيير أبناط الحروف.
- ✓ تحاشى الجمل الطويلة والمعقدة والمصطلحات المهجورة والاختصارات والمرادفات.
- ✓ راعى مستوى المتعلم من حيث مستوى الكتابة وحصيلة مفرداته اللغوية .
- ✓ استخدم المصطلحات بشكل موحد ومتناسق على امتداد الدرس وبالتالي المنهج.
- ✓ راعى تسلسل العرض ومنطقياته من خلال التمهيد والتركيز على الجوهر ، وترك التفاصيل التي تشتت الطالب ، والانتقال من الأبسط فالأعقد.
- ✓ استخدم أساليب التماثل ، والربط بين ما يعرض وما سبق عرضة ، وارجع ما يجرى تقديمه إلى أنماط ونماذج تسهل على المتعلم وتسرع من استيعابه.
- ✓ استخدم القوائم وجداول العلاقات ومتدرجاتها وذلك لإبراز علاقات الترابط بين العناصر المختلفة.
- ✓ لا تسرد شرحا مطولا ؛ يتطلب من المتعلم التدخل من جانبه .
- ✓ استخدم الألوان دون أخطاء وظيفية كلما أمكن – الماء : أزرق ، الدم : أحمر ، الرمل : أصفر ، والشجر : أخضر.

- ✓ استخدم الأشكال والمؤثرات الصوتية ولقطات الفيديو والكرتون كلما أمكن ذلك ، ولكن دون مبالغة.

- ✓ لا تطلب من المتعلم أن يفعل شيئين في وقت واحد ؛ كن واضحا وقاطعا في رسائلك ولا تستخدم تلك ذات الطابع الشفهي مثل : ما هو رأيك ؟ ، أود أن اسمع إجابتك؟

- ✓ لا تجعل فترة تلقى الدرس قصيرة للغاية أو طويلة للغاية .

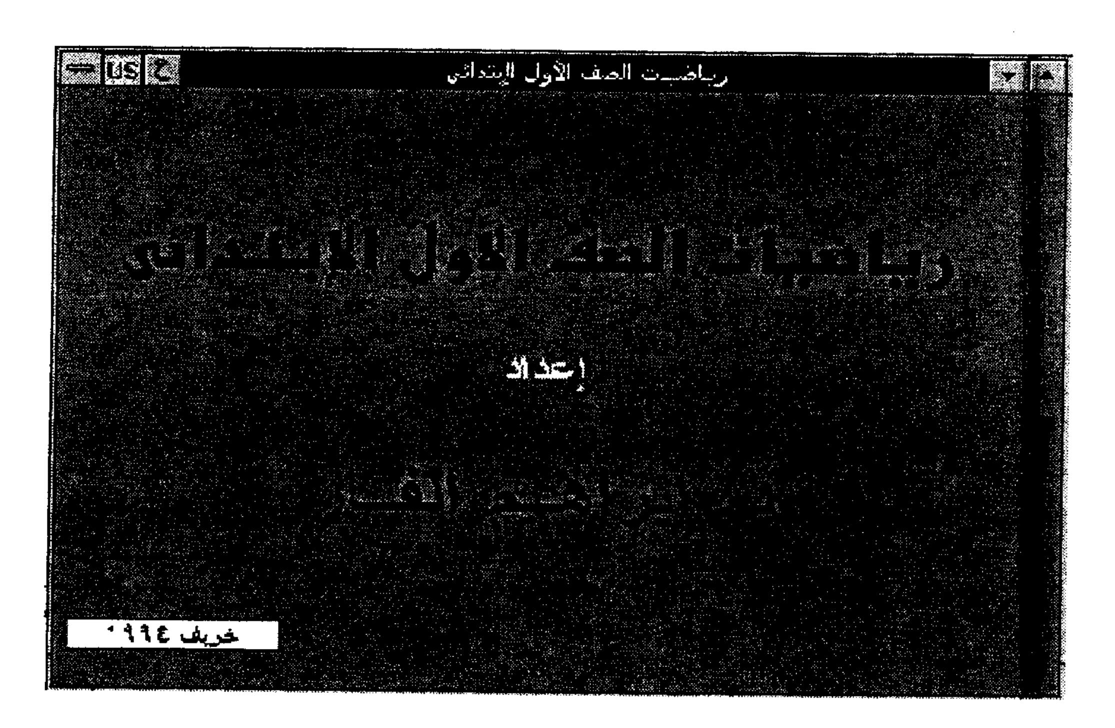
- ✓ تجنب الانتقال السريع من شاشة إلى أخري أثناء العرض لإعطاء فرصة كافية للمتعلم للقراءة والتفكير والاستجابة ، مع تجنب البطيء الشديد الذي يولد لدى المتعلم الملل والضجر.

- ✓ راعي تنويع أساليب التغذية الراجعة وتنوعها : رسالة ، صوت ، رسوم ، أشكال ، رسوم متحركة ، لقطة فيديو ، قطعة من أغنية مشهورة ومحبوبة ، السلام الوطني ، .... ؛ وذلك بالنسبة للإجابة الصحيحة والخاطئة على حد سواء.

تصنيف شاشات البرمجية

ويمكن تصنيف شاشات البرمجية بصفة عامة إلى الأنواع الأساسية التالية :

١- **شاشة التعريف بالبرمجية :** وعادة ما توضح اسم المنفذ والمصمم والمراجع والمشرف والسينياريست ... الخ ، ويعتبر النموذج التالي ابسط شاشات التعريف بالبرمجية :



شكل رقم (٧)

شاشة تعريف بالبرمجية
تحتوي بعض النصوص



شكل رقم (٨)

شاشة تعريف بالبرمجية
تحتوي بعض النصوص والصور

ويعتر هذا النموذج لشاشة التعريف بالبرمجية أكثر تطورا من النمـــوذج السابق لاحتوائية علي صورة وبعض الكلمات الواعدة للمستخدم مثل التعليـــم والتعلم المعزز بالحاسوب متعة ، وكذا تنوع أبناط الخطوط وأحجامها.



شكل رقم (٩)

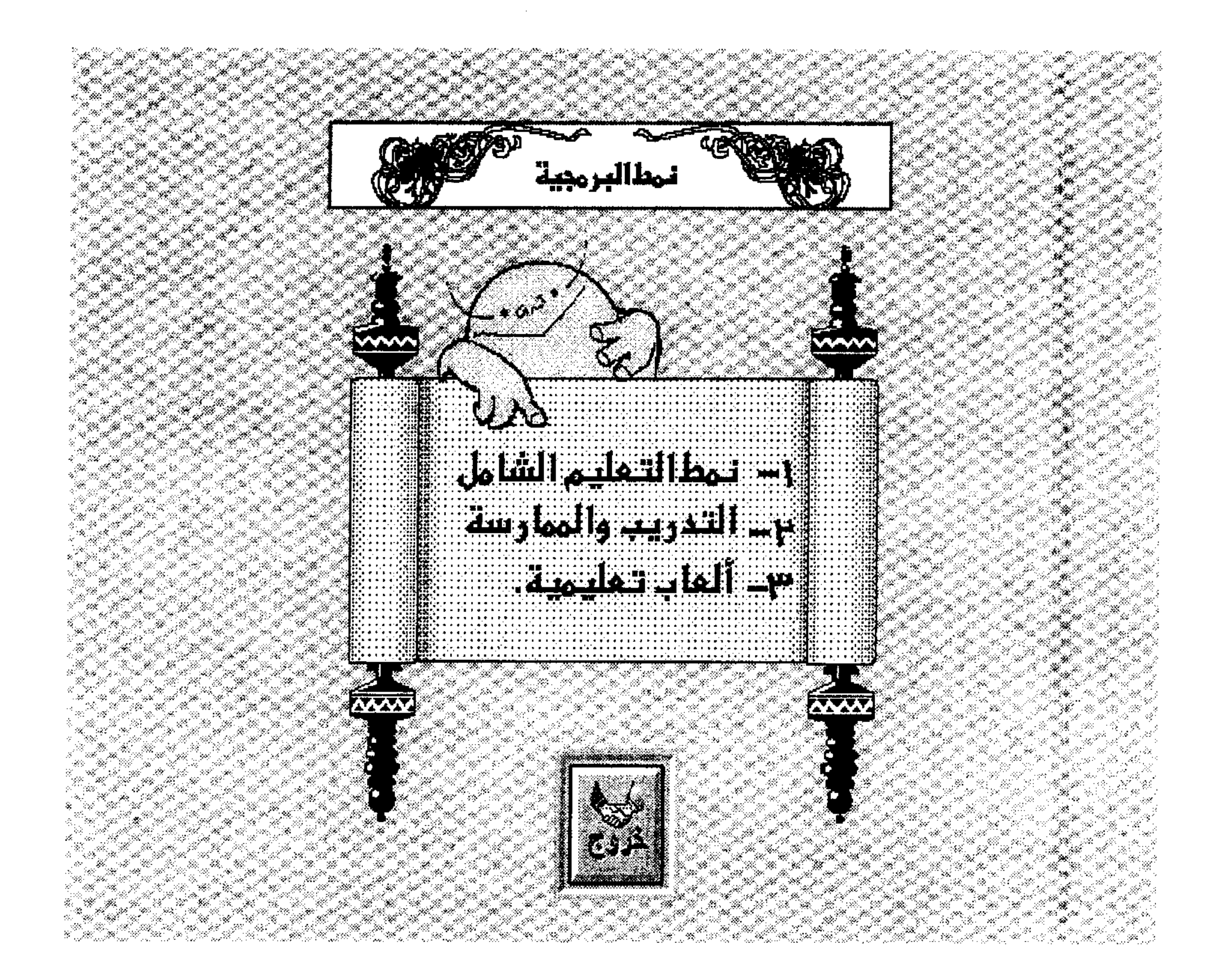
شاشة تعريف بالبرمجية
تحتوي بعض النصوص والصور
وإطار متميز

ويعتبر هذا النموذج أكثر تطورا من النماذج السابقة ، حيث أن النصوص التي احتوتها الشاشة هي نصوص متحركة ، تظهر وتختفي بالتناوب مع الشاشة الموضحة بالشكل رقم (٢١) ، يصاحب هذه الحركة التعريف الصوتي بالبرمجية .



شكل رقم (١٠)

شاشة تعريف بالبرمجية
ضمن التسلسل السابق لشاشات
التعريف بالبرمجية



شكل رقم (١١)

شاشة مكملة
لشاشات التعريف بالبرمجية

والذي يوضح ماتحتوية البرمجية من أنشطة مثل : **التعليم الشامل** و**التدريب والممارسة** و**الألعاب التعليمية** .



شكل رقم (١٢)

شاشة تعريف بالبرمجية
تحتوي بعض النصوص والصور

واضح أن هذه الشاشة تعرف المتعلم بالبرمجية بصورة واضحة وشاملة، حيث تتضمن المعدين والمشرف وموضوع البرمجيـــة ، إضافـــة لاحتوائهـــا بعض الصور المتحركة ، وكذا التعبيرات الصوتية المهيئة للمتعلم.

٢- شاشة المقدمة : والتي تتكون عادة من شاشة واحدة أو أكثر ، حيث يتم من خلال شاشات هذا الجزء:

- التعريف بموضوع البرمجية
- تشويق المتعلم للبرمجية وموضوعها



شكل رقم (١٣)

نموذج لشاشة مقدمة توحي بالحركة عن طريق النص

يحتوى هذا النموذج علي حركة متوافقة للنص **رياضيات الصف الأول** يساعد على لفت نظر المتعلم مما قد يحفزه علي مداومة التعلم من خلال البرمجية .



شكل رقم (١٤)

نموذج لشاشة مقدمة أخرى
توحي بمزيد من الحركة
بإستخدام بنل مع بعض الصور

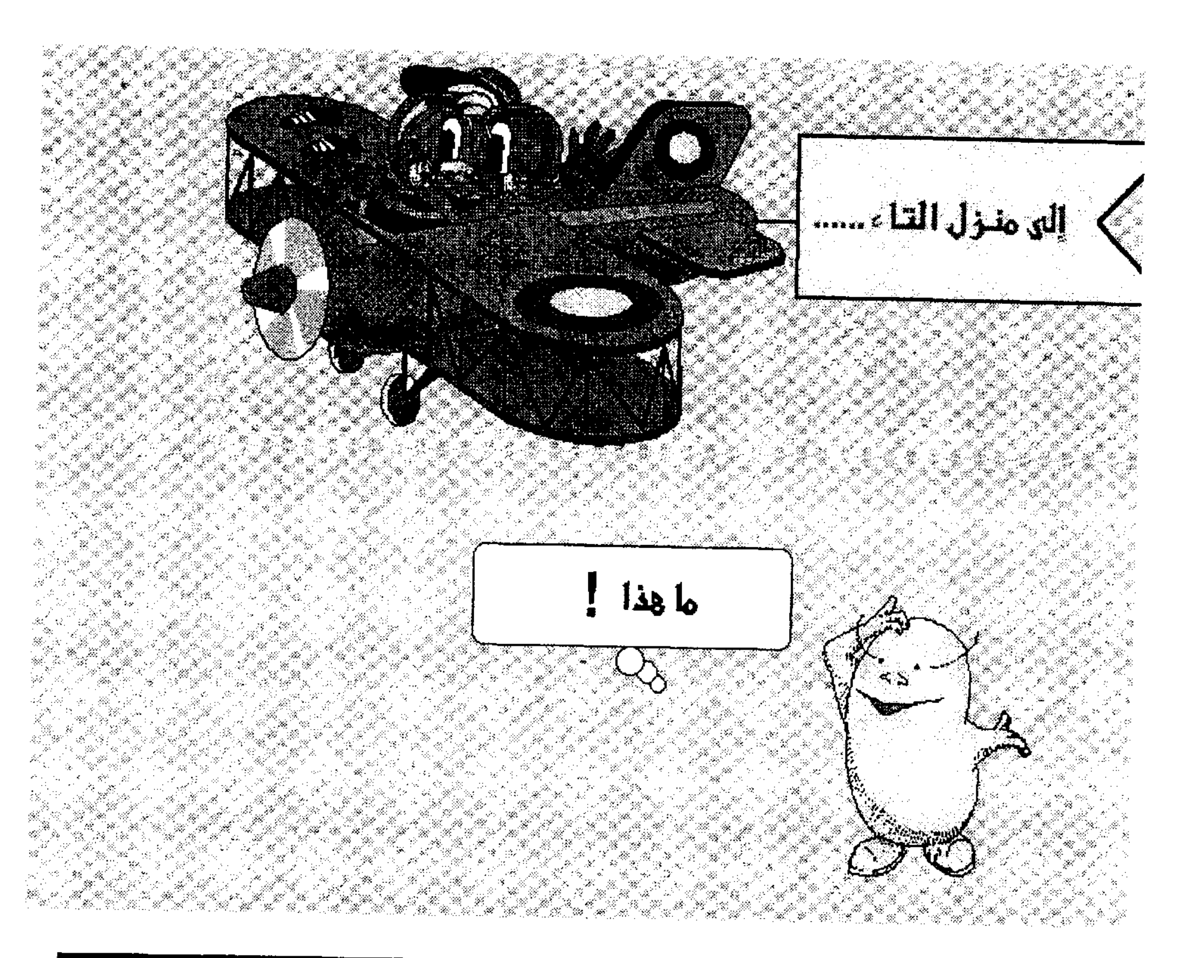
تحتوي هذه الشاشة على زر ، عندما يضغط عليه المتعلم ينتقل إلى الشاشة التالية ، أما في حالة عدم الضغط تتوالى الحركات المتناسقة والدائمة لمكونات الشاشة مع تأثير صوتي جذاب .



شكل رقم (١٥)

نموذج لشاشة مقدمة
تتضمن نصوص وصور

يصاحب ظهور هذه الشاشة قراءة لبعض أبيات الشعر العربي الذي يوضح جمال لغتنا العربية ، مع موسيقى تصويرية عذبة ، إضافة إلى حركة تمايل الزهور وتغيير ألوان ظهورها.



شكل رقم (١٦)

نموذج لشاشة مقدمة

يتضمن هذا النموذج حركة الطائرة إلى **منزل التاء** ، مما حذا بالشكل التعبيري (كراكتيري) المتحرك إلى التعجب من هذه الحركة ، والذي سوف يقرر مع التلميذ فيما بعد إلى اللحاق بالمتعلمين المستقلين بالطائرة والذهاب إلى منزل التاء.

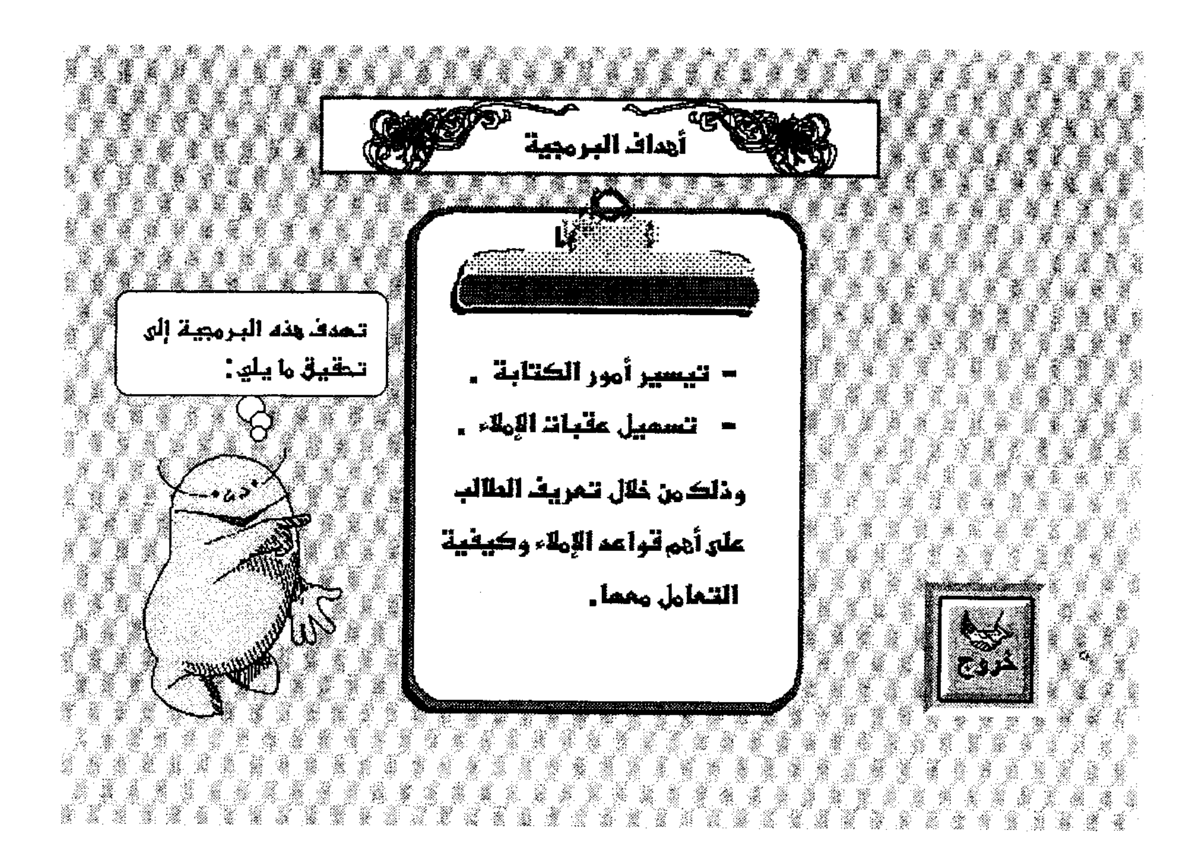


شكل رقم (١٧)

نموذج لشاشة مقدمة

يتضمن هذا النموذج علي حركات متناسقة للنصوص والشكل التعبـيري ، مع سماع أصوات معبرة ، تساعد علي تهيئة بيئة التعلم للتلاميذ.

٣- شاشة الأهداف : والتي تتكون عادة من شاشة واحدة أو أكثر ، حيث تتضمن علي الأهداف العامة للبرمجية والأهداف السلوكية لأداء الطالب .



شكل رقم (١٨)

نموذج لشاشة أهداف عامة

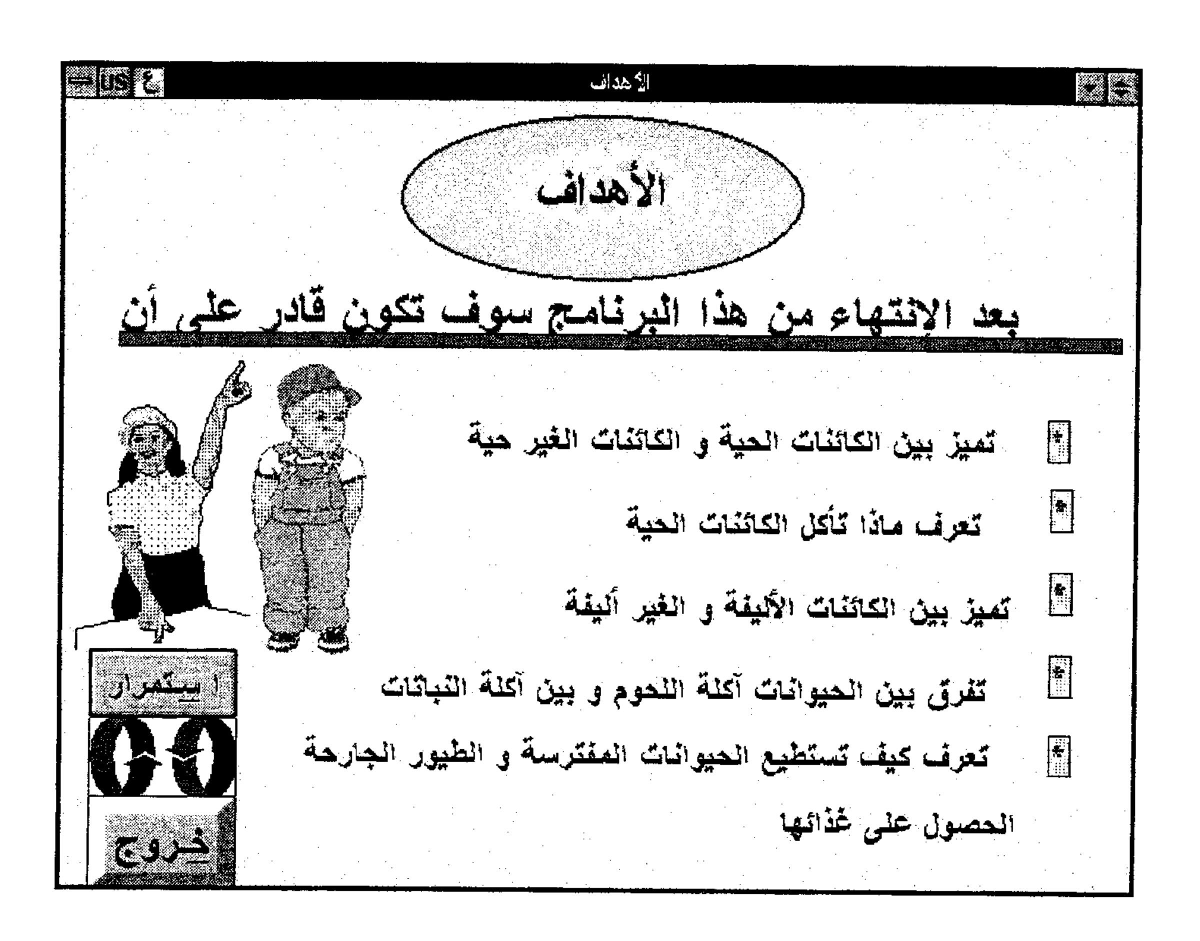
لاحظ مكونات الشاشة من رسومات وزخارف ونصوص ، علماً بأن الشكل التعبيري يقوم بقراءة الأهداف بصوت واضح .



شكل رقم (١٩)

نموذج لشاشة أهداف سلوكية

لاحظ أن هذه الأهداف مرحلية ، بمعني أنها أهداف زيارة منزل التاء فقط، وقد يكون هذا افضل بدلا من سرد جميع أهداف دروس البرمجية كاملة مرة واحدة.



شكل رقم (٢٠)

نموذج لشاشة أهداف سلوكية

صورة أخري لكتابة الأهداف ، مع ملاحظة أن الشاشة تتيح الحرية كاملة للمتعلم في التوقف عند هذا الحد أو الاستمرار في التعلم من خلال البرمجية .

٤- **شاشة القائمة** : والتي تتكون عادة من شاشة واحدة أو أكثر ، ، وهي التي يمكن عن طريقها أن يتحكم المتعلم في سير الدرس ، وعادة ما تتكون من مجموعة من الأزرار *free menu* أو القوائم المنسدلة *drop down menu* ، حيث يتم عن طريق اختياراتها تدفق بقية شاشات البرمجية ، هذا وتعتبر هذه الشاشة من ناحية أخرى مرآة تعكس ما تحتويه البرمجية من دروس وموضوعات ، وعادة ما تشتمل علي الخيارات الرئيسية التالية :

- الأهداف الخاصة بالدرس .
- عرض الدرس .
- تقويم أداء المتعلم .
- ملخص لموضوعات أو فقرات الدرس .
- إمكانية الخروج من البرنامج .
- إمكانية تعريف المتعلم بكيفية التعامل مع البرمجية .
- التعريف بموضوع البرمجية .

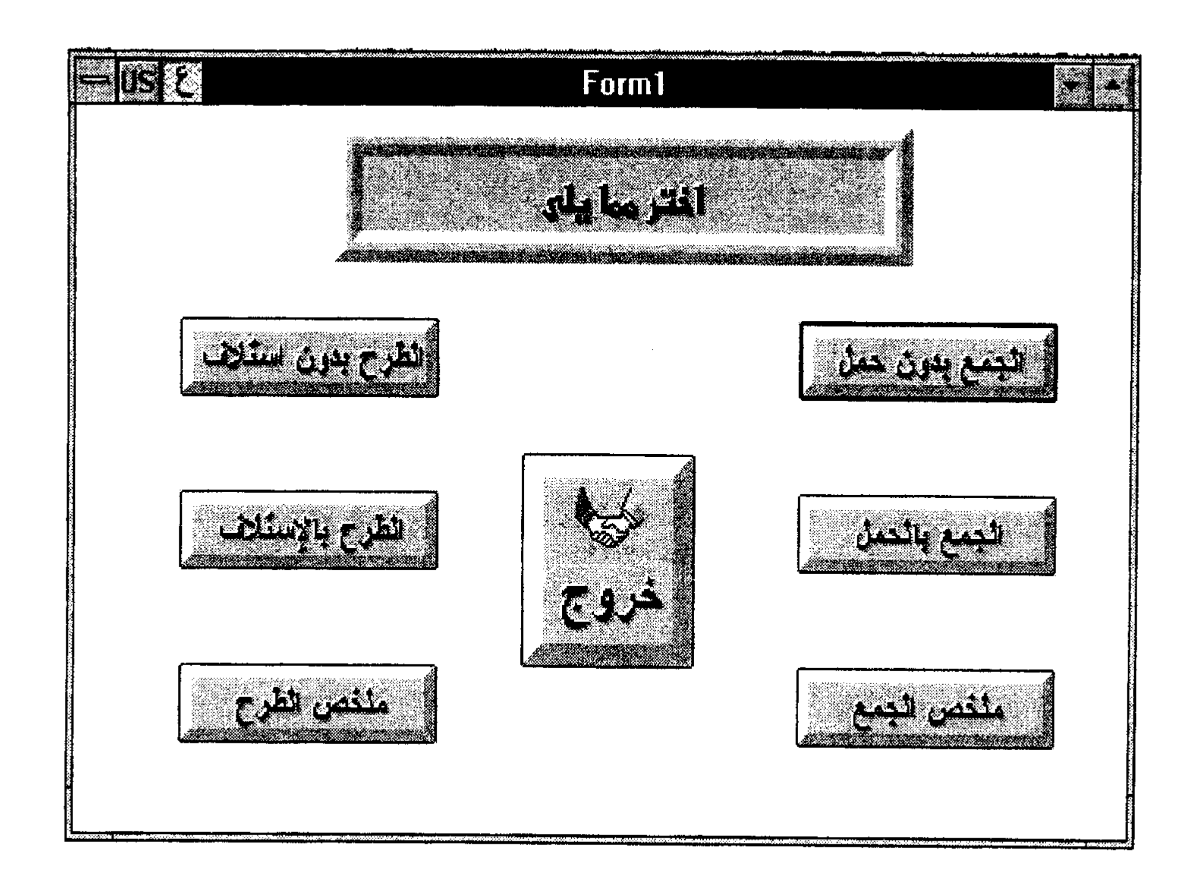
والتي غالبا ما يتفرع منها العديد من الخيارات الفرعية *sub menu* الأخرى .

مثال (٢) : تصميم شريط قوائم

الجمع　الطرح　الخلاصة　خروج

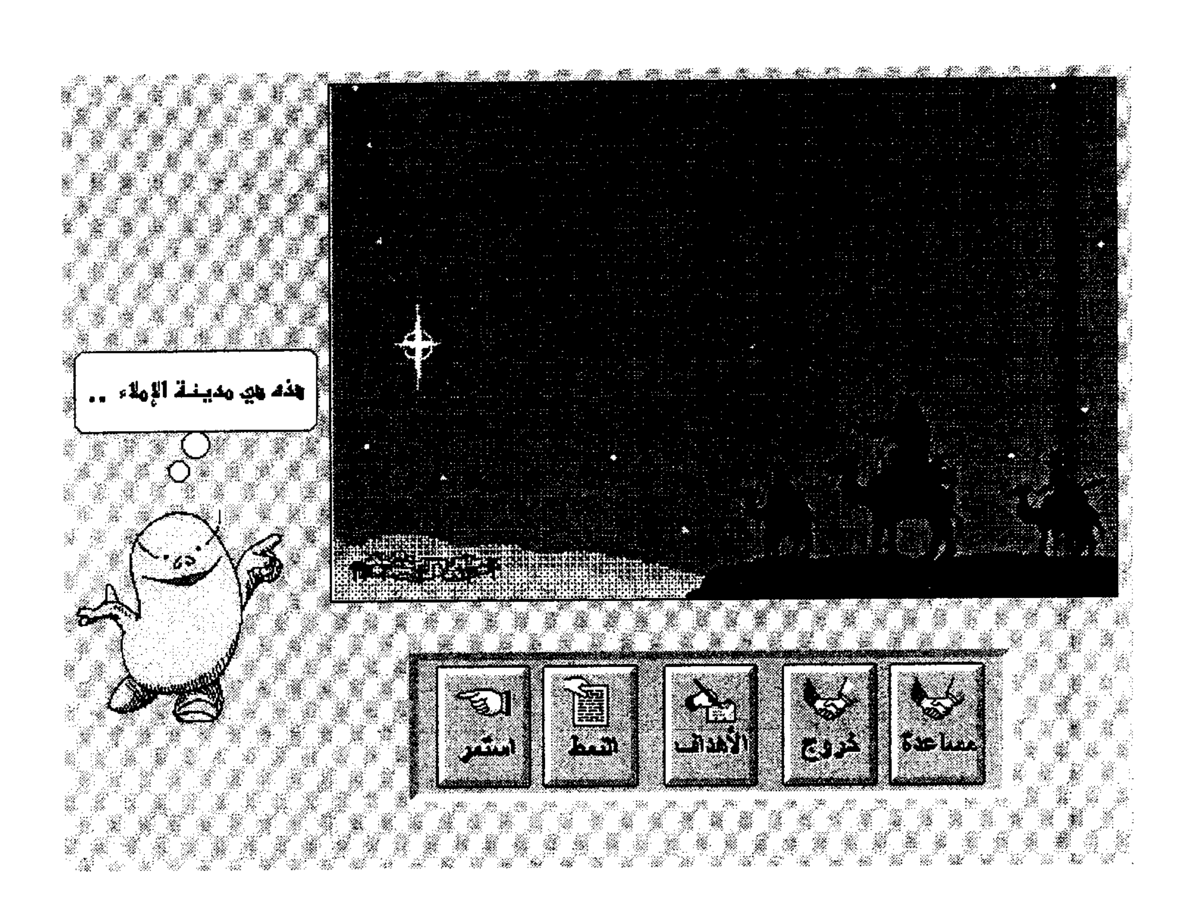
شكل (٢١)

شاشة قائمة
موضحا عليها شريط قوائم منسدلة



شكل (٢٢)
شاشة قائمة حرة
عليها ازرة للموضوعات كقائمة حرة

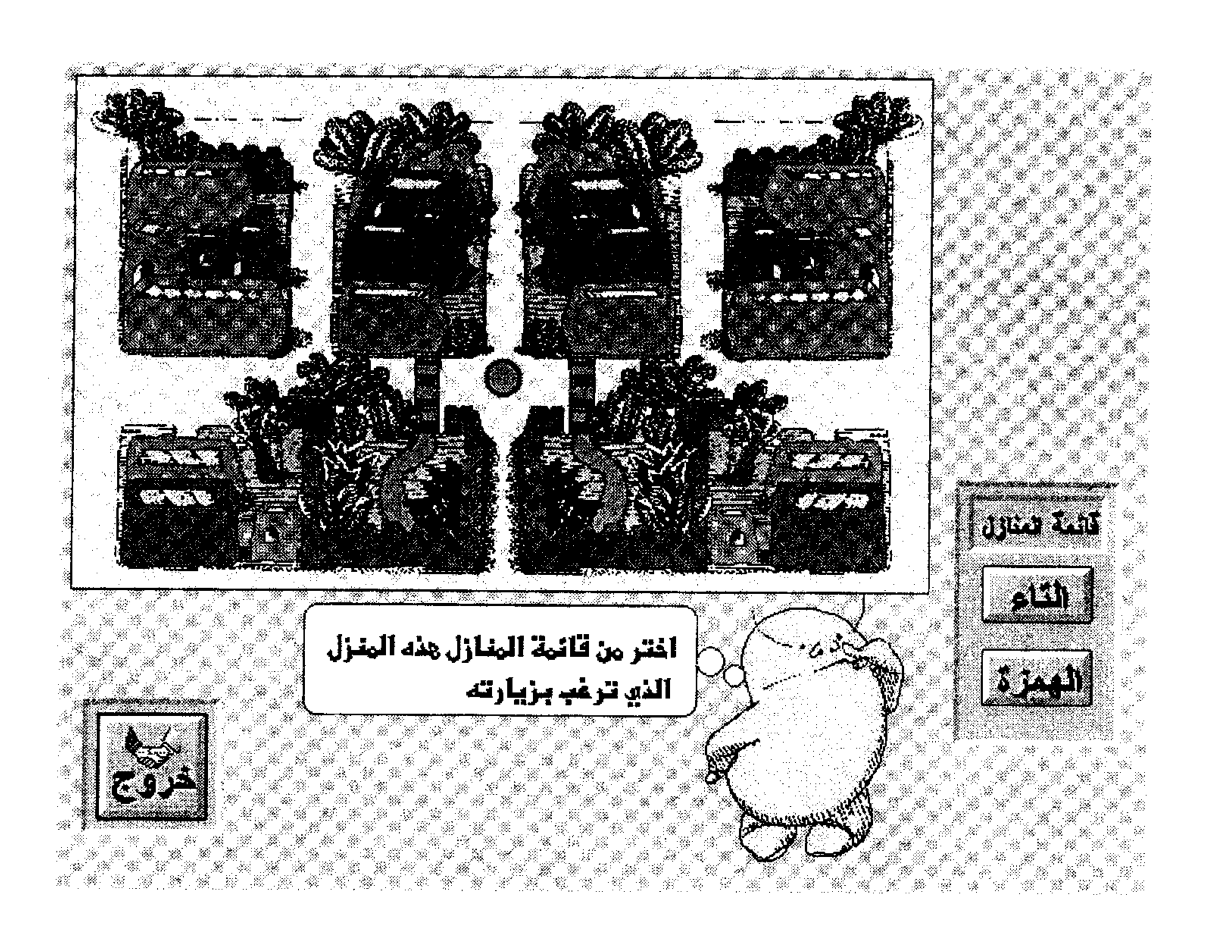
إن كل اختيار تتضمنه هذه الشاشة يتفرع بالطبع إلى اختيارات (أو قوائم) فرعية أخري تتضمن فقرات كل موضوع .



شكل رقم (٢٣)

نموذج لشاشة قائمة حرة

لاحظ صورة الخلفية ، ومفاتيح التفاعل ، والتي تتيح الفرصة للمتعلـــم أن يحصل علي مساعدة عن كيفية الاستخدام قبل بدأ تعلمه من خلال البرمجية.



شكل رقم (٢٤)

نموذج لشاشة قائمة جزئية حرة

لاحظ أن تلك الشاشة هي شاشة قائمة فرعية لأحد الدروس ، وهو درس **التاء والهمزة** ، وليست قائمة عامة كما في شكل (٢٣)



شكل رقم (٢٥)

نموذج لشاشة قائمة شاملة حرة

واضح أن تلك الشاشة تتضمن قائمة بكافة موضوعات البرمجية ، مع خيار خروج المتعلم من البرمجية حالة رغبته في ذلك ، إضافة إلى أن كل من الخيارات السابقة تتفرع إلى قوائم فرعية لمحتويات الموضوعات.



شكل رقم (٢٦)

نموذج لشاشة قائمة
لاختبارات الدروس البنائية

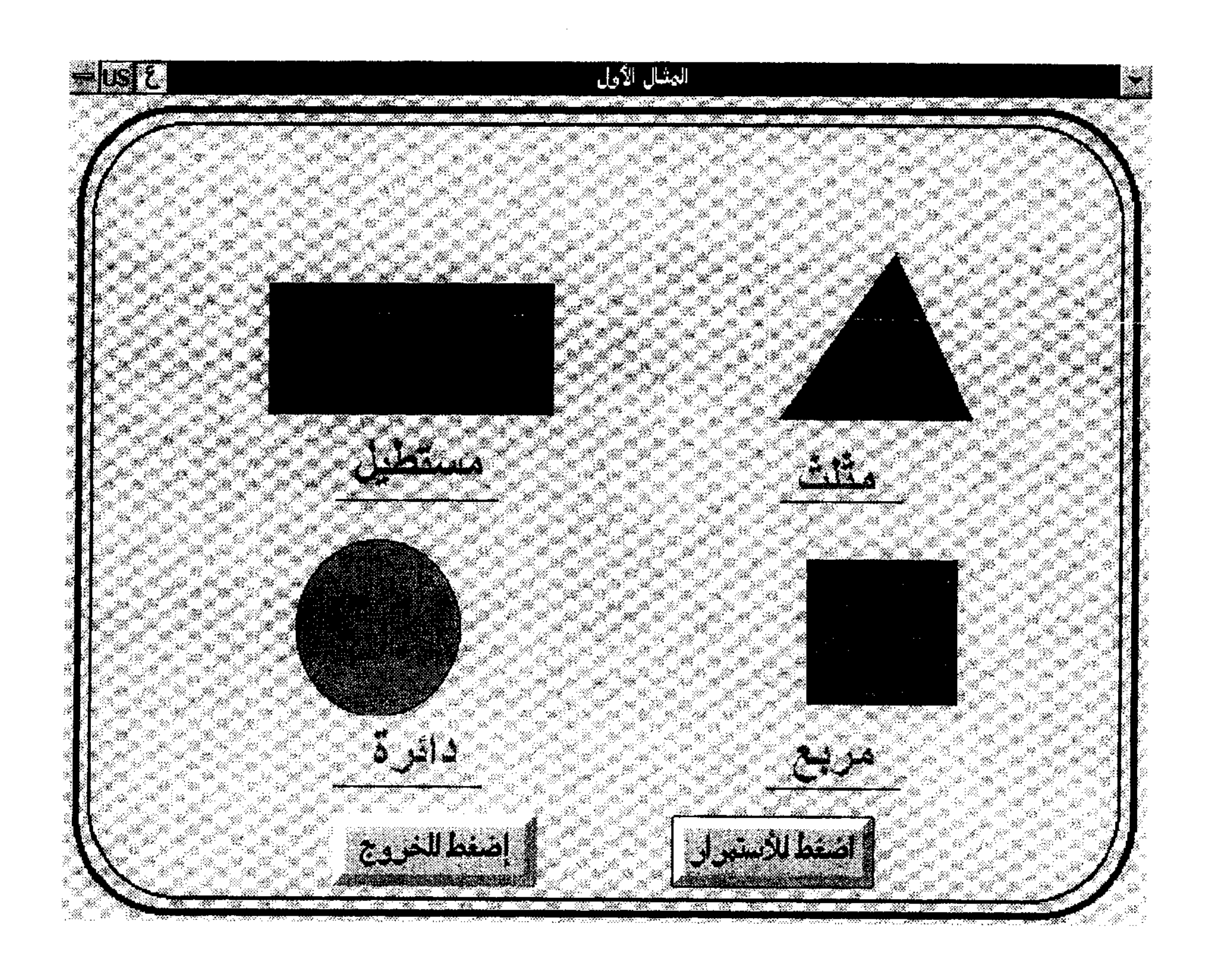
٥- **شاشة العرض** : وعادة ما تكون مجموعة متنوعة من الشاشات ، والتي يتم من خلالها عرض المواد التعليمية من خلال المشاركة الفعالة من قبل المتعلم ، مع إتاحة الفرصة له للقيام بالعديد من الأنشطة الفردية : الذاتية والتعاونية ، مع ضرورة احتوائها علي بعض الصور الثابتة والمتحركة والرسوم التوضيحية ، ولقطات الفيديو اللازمة للشرح والتوضيح ، وإتاحة الفرصة للاستيعاب النشط والفعال.



شكل رقم (٢٧)

نموذج لشاشة عرض

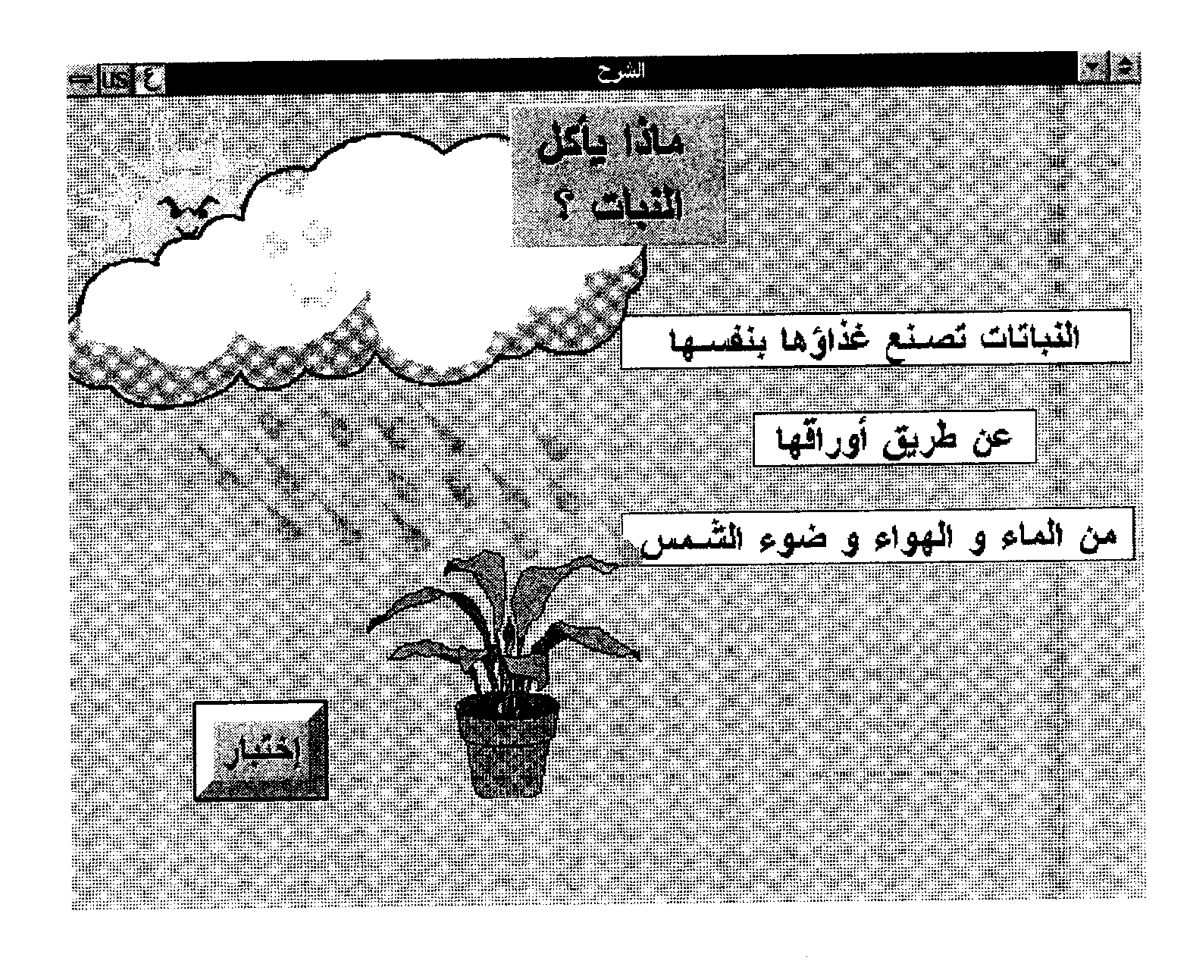
نموذج بسيط لشاشة عرض ، ولكن مصدر تميزه يكمن في المكونات اللونية للشاشة وكذا تصميمها.



شكل رقم (٢٨)

نموذج لشاشة عرض تفاعلية

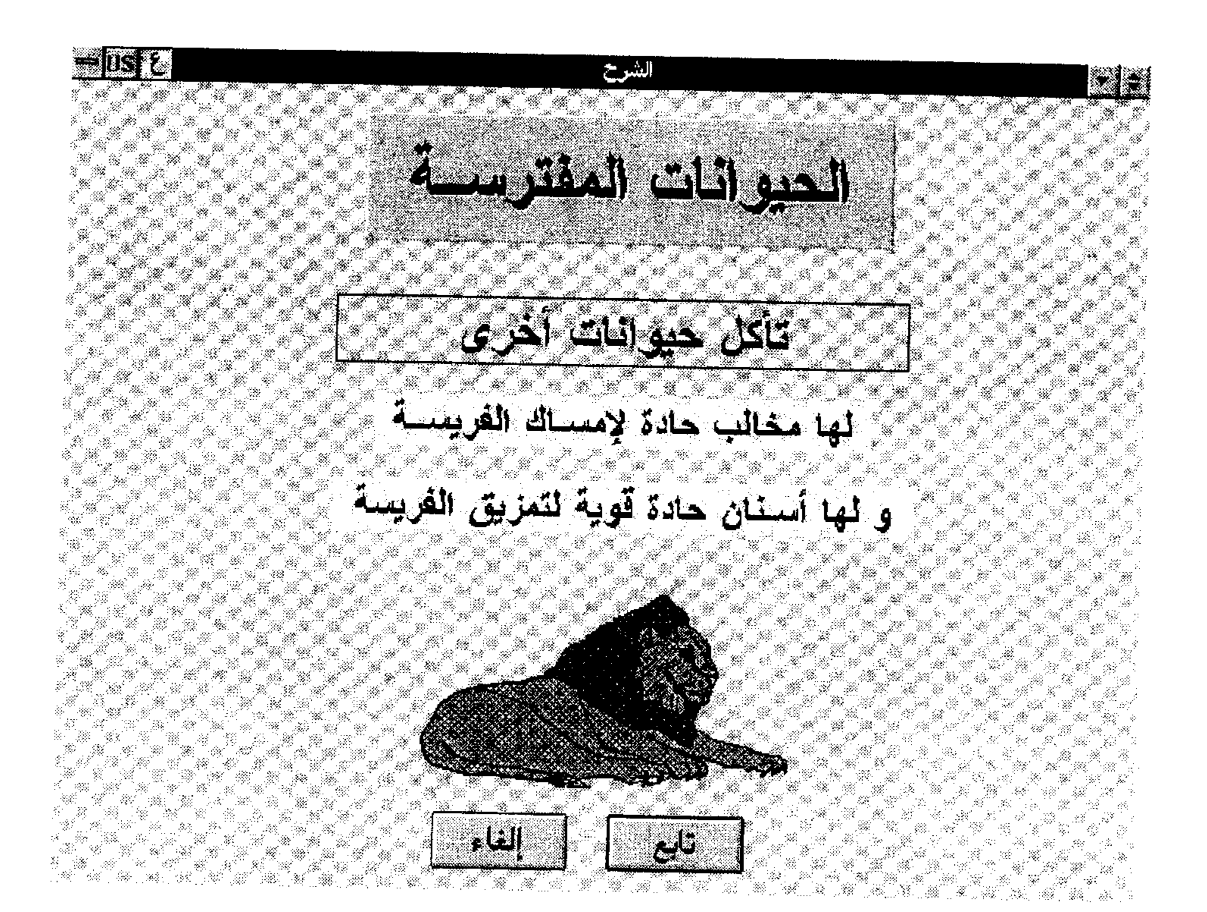
يتيح هذا النموذج للمتعلم أن يتدرب علي نطق مسميات الأشكال المتضمنة بالشاشة ، حيث يتعرف المتعلم علي الأشكال بسماع مسميات‍ها كلما أشر بالماوس علي أي منها.



شكل رقم (٢٩)

نموذج لشاشة عرض

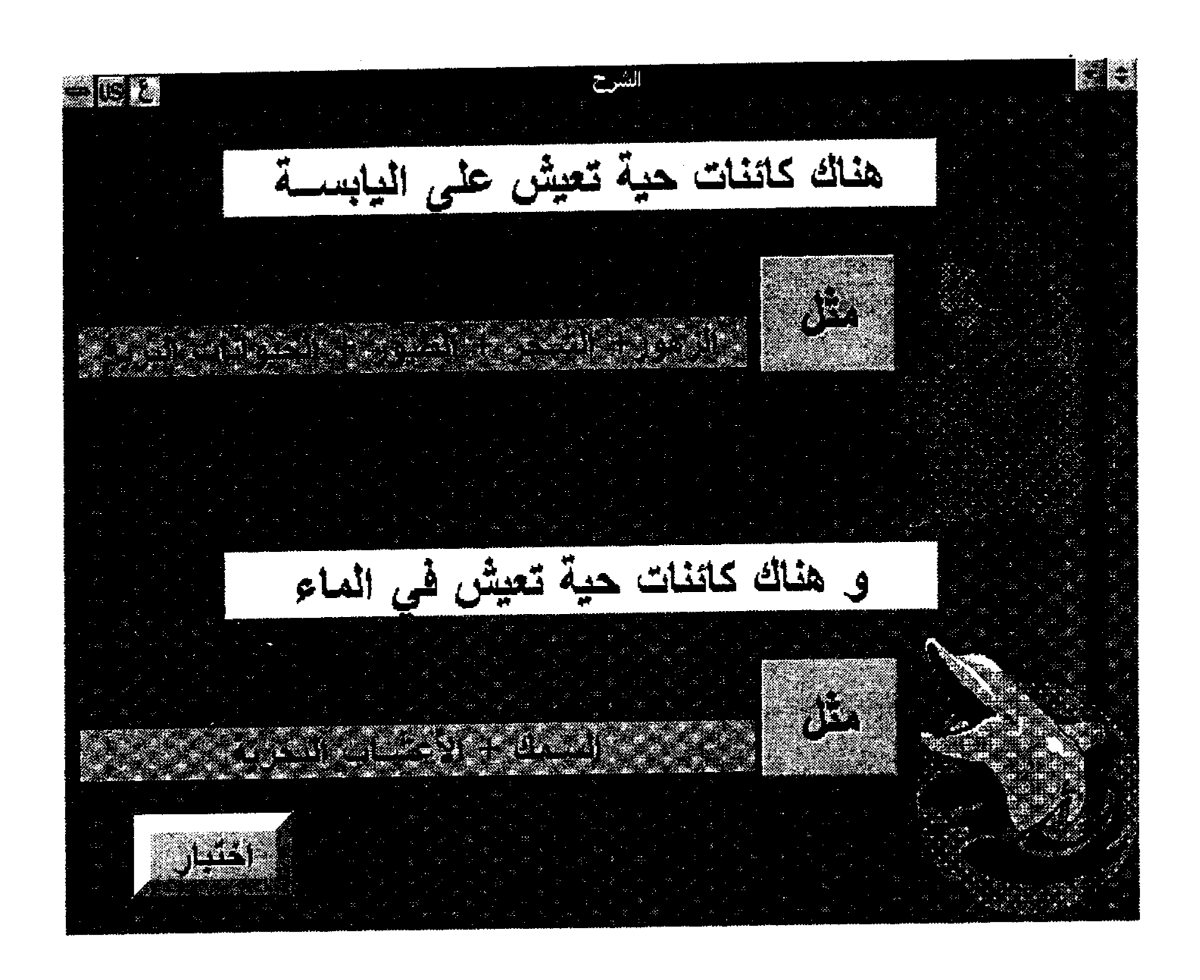
لاحظ الرسم المعبر ، والذي يساعد المتعلم علي فهم مكونات غذاء النباتات ، وكيفية صنعه ، حيث تقوم النباتات بصنع غذائها بنفسها عن طريق أوراقها من الماء والهواء وضوء الشمس.



شكل رقم (٣٠)

نموذج لشاشة عرض

لاحظ احتواء هذا النموذج علي مواصفات الحيوانات المفترسة مع وجود مثال لتلك الحيوانات **الأسد** .

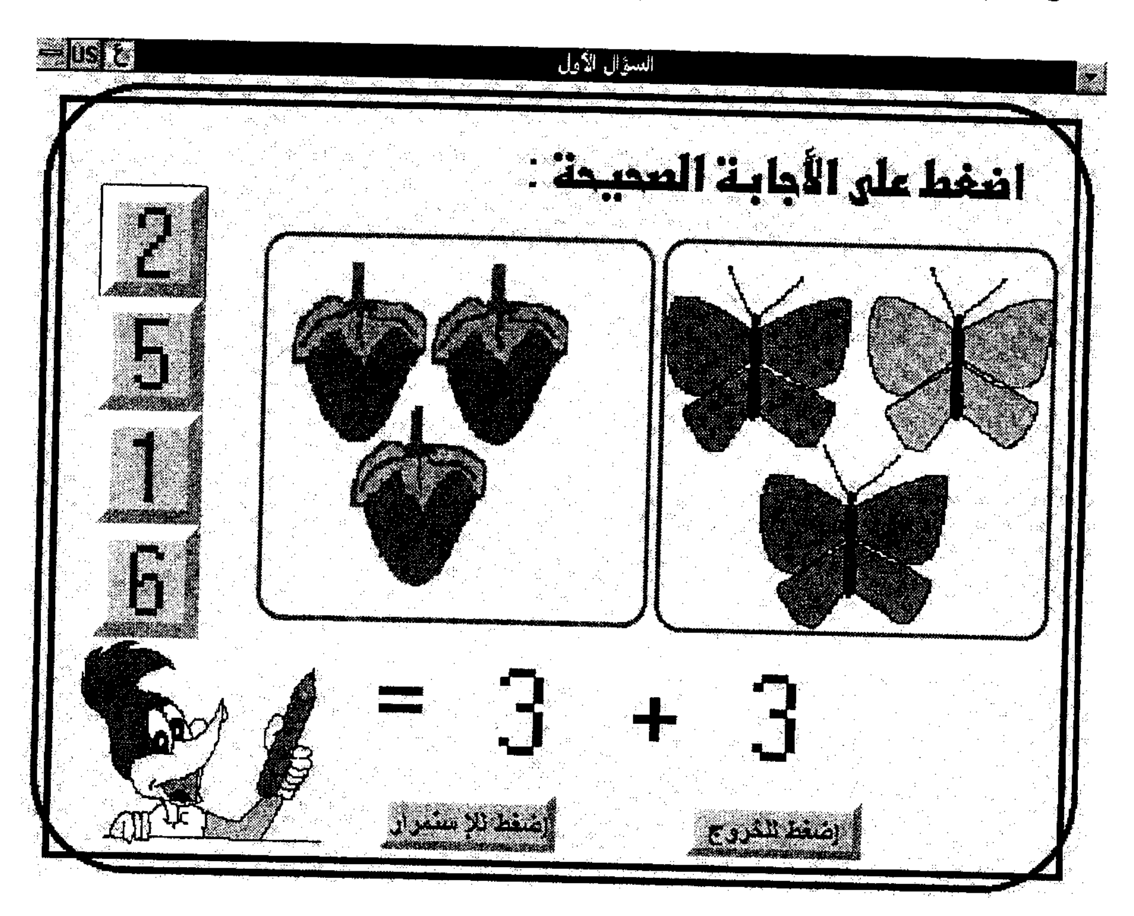


شكل رقم (٣١)

نموذج لشاشة عرض بأمثلة

تعتبر شاشة العرض بأمثلة أحد نماذج شاشـــات العـــرض ، حيـــث يتـــم العرض بأمثلة غير تفاعلية ، ويوضح هذا النموذج أمثلة لأنواع الحيوانـــــات التي تعيش علي اليابسة مع إعطاء أمثلة لها . يوجد زر لبدء أخذ اختبـــــارات بسيطة تطبيقية لهذا النوع من الحيوانات.

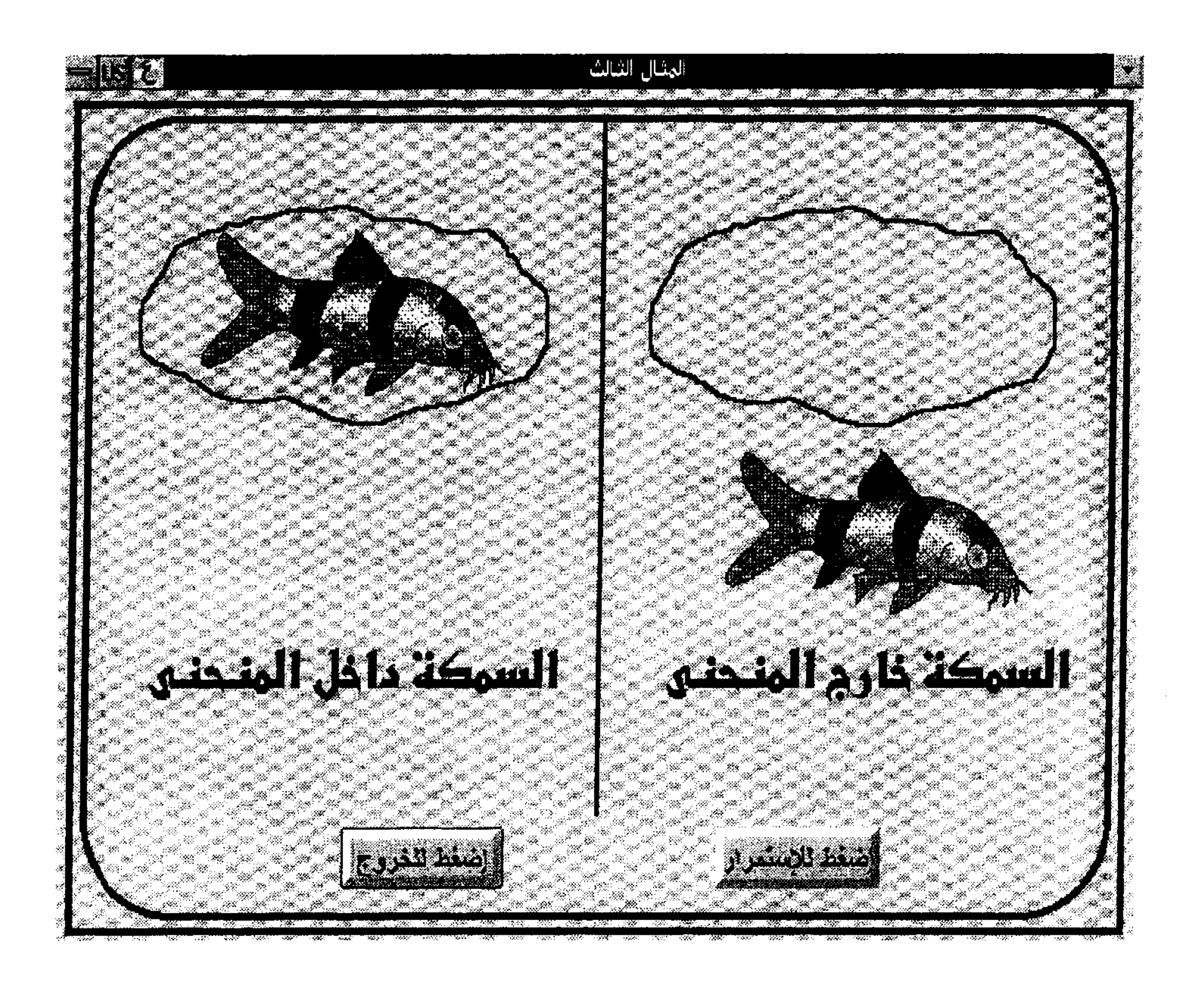
٦- **شاشة الأمثلة والتمارين والتدريبات :** وعادة ما تكون مجموعة متنوعة من الشاشات ، والتي يتم من خلالها عرض الأمثلة والتمارين والتدريبات المتنوعة ، مع ضرورة التركيز علي تنوع التغذية الراجعة ؛ والتي تتضمن عادة : رسالة ، صوت ، رسوم ، أشكال ، رسوم متحركة ، لقطة فيديو ، قطعة من أغنية مشهورة ومحبوبة ، السلام الوطني ، .... ؛ علي أن تكون معبرة للإجابة الصحيحة والخاطئة على حد سواء.



شكل رقم (٣٢)

نموذج لشلشة أمثلة
(علوم)

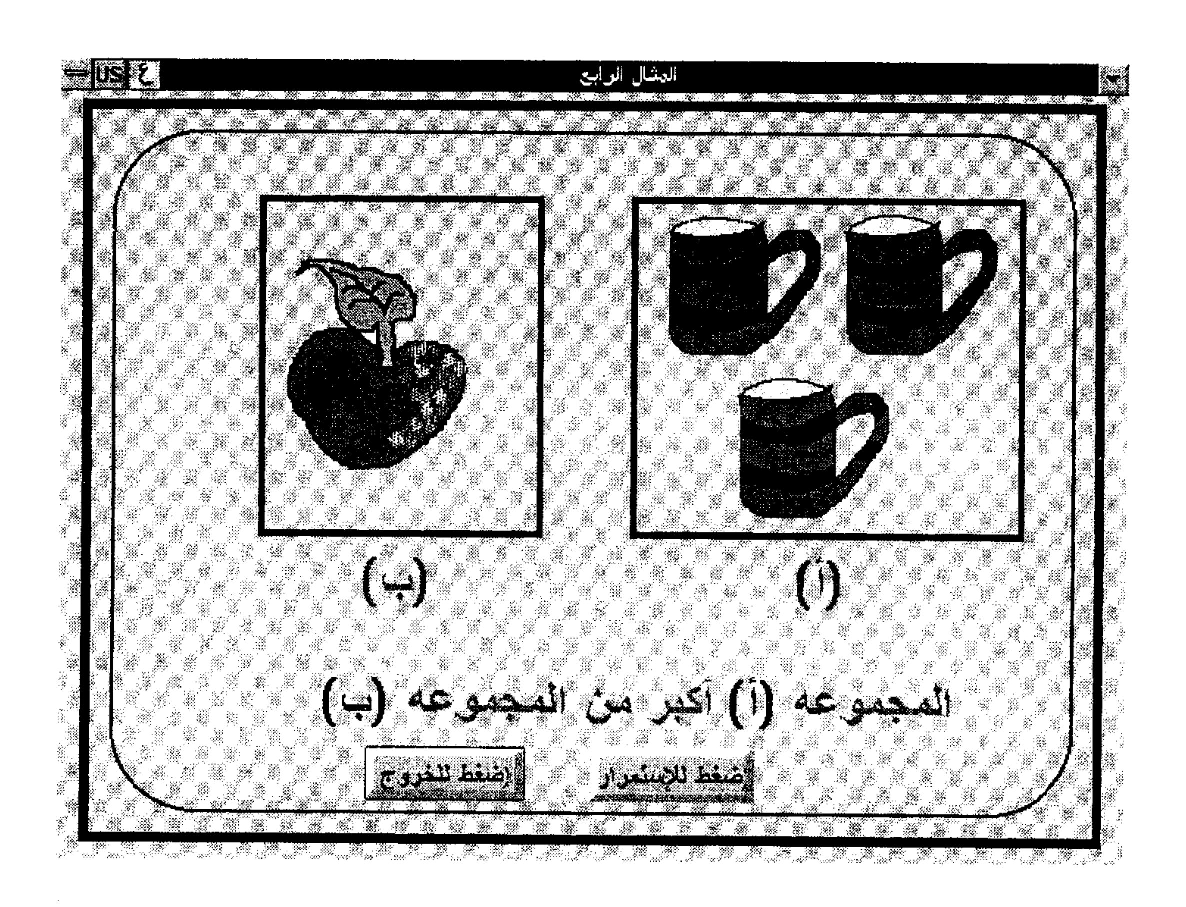
لاحظ التصميم الجيد لهذا النموذج لأحد شاشات الأمثلة التفاعلية ، والتي تتطلب من المتعلم التفاعل والمشاركة في الحل.



شكل رقم (٣٣)

نموذج لشاشة أمثلة
(رياضيات)

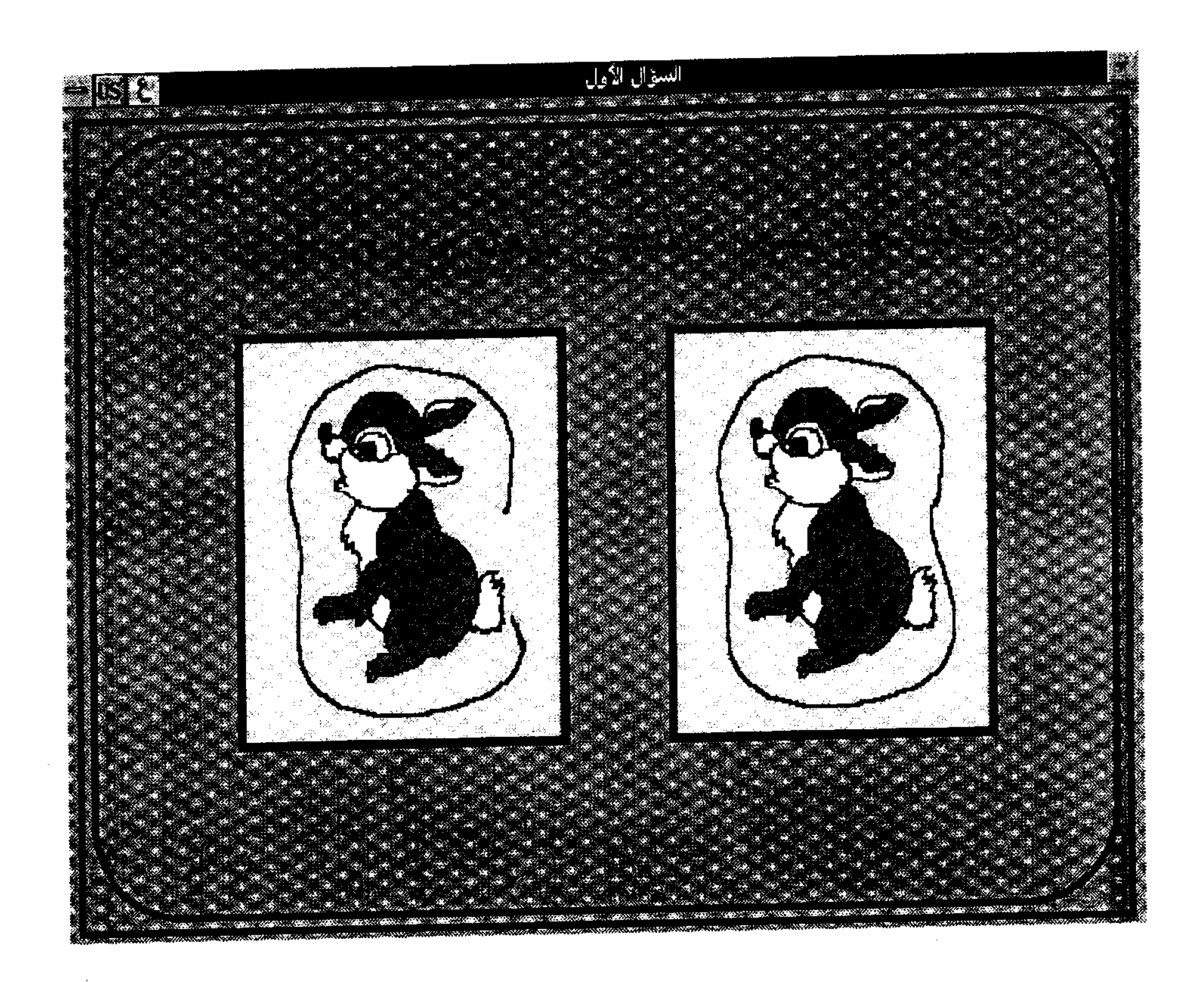
لاحظ تصميم هذا النموذج لأحد شاشات الأمثلة الغير تفاعلية ، والتـــي لا تتطلب من المتعلم أي نوع من التفاعل أو المشاركة في الحل.



شكل رقم (٣٤)

نموذج لشاشة أمثلة
(رياضيات)

وهو عبارة عن نموذج آخر كالنموذج السابق ؛ نموذج لشاشة أمثلة غير تفاعلية والتي لا تتطلب من المتعلم أي نوع من التفاعل أو المشاركة.

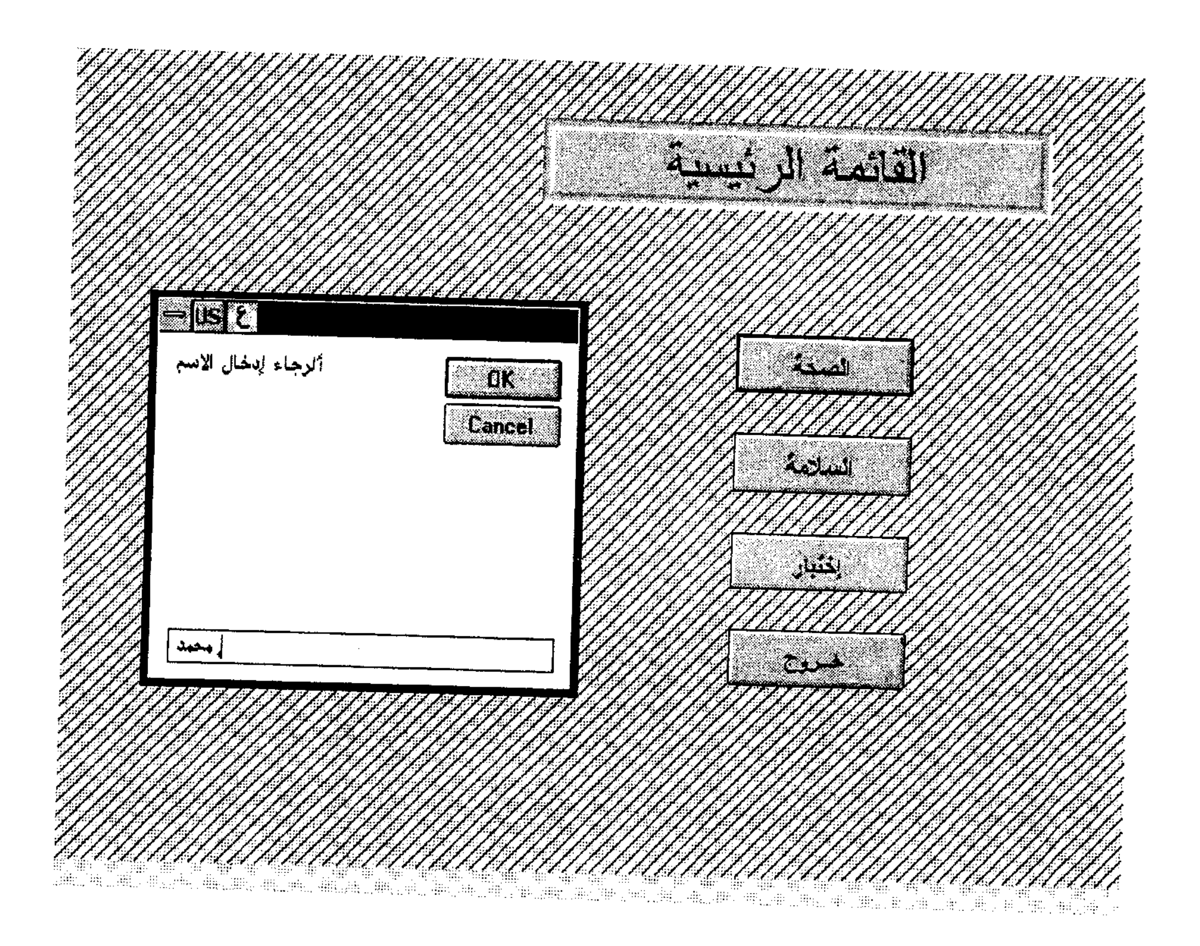


شكل رقم (٣٥)

نموذج لشاشة أمثلة
(رياضيات)

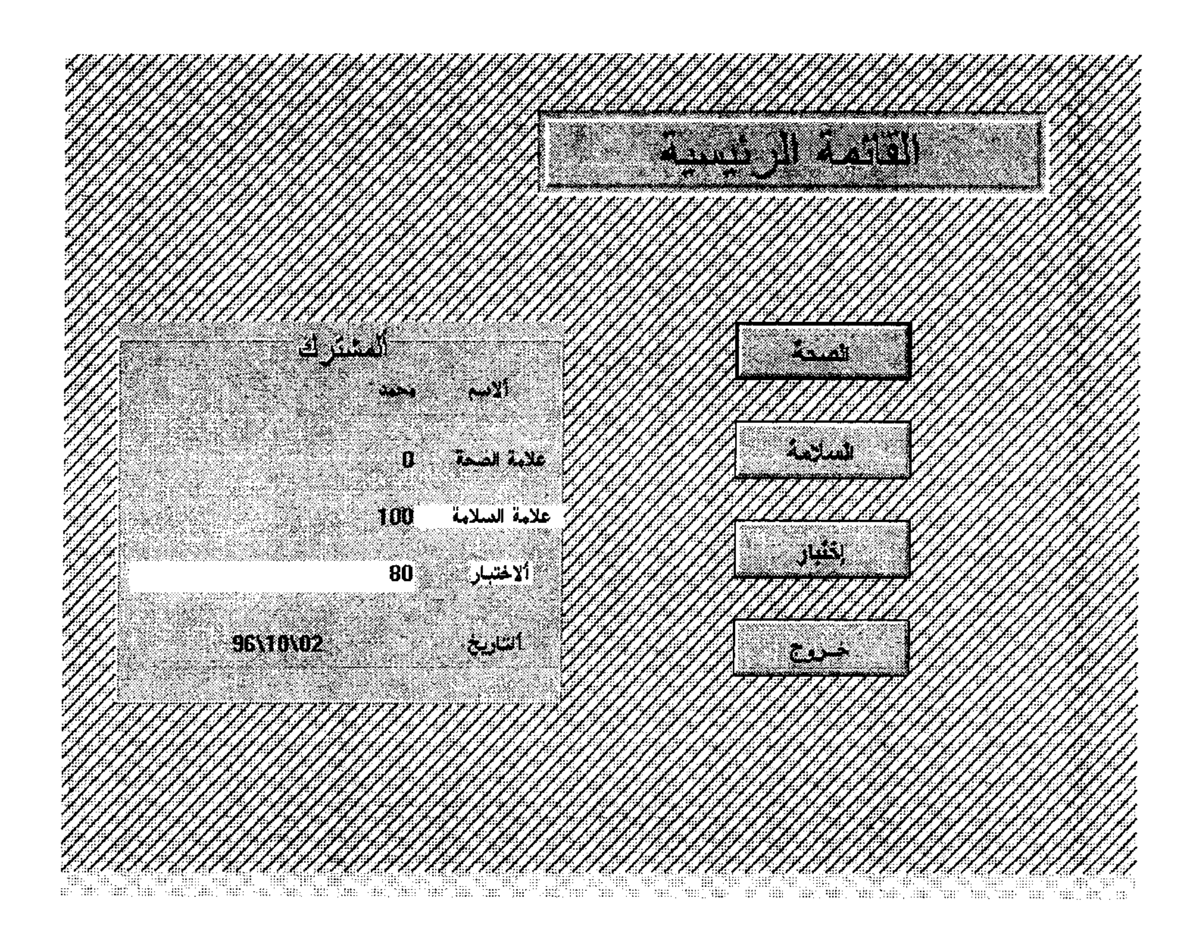
لاحظ تصميم هذا النموذج لأحد شاشات الأمثلة التفاعلية ، والتي تتطلب من المتعلم التفاعل والمشاركة في الحل.

٧- **شاشة التقويم :** وعادة ما تكون مجموعة متنوعة من الشاشات ، والتـــي يتم من خلالها عرض مفردات الاختبار بــــهدف التقويــم أو التشـــخيص أو الإتقان، والتي ينبغي أن تكون متنوعة قدر المستطاع وحسب نوع الاختبـــار المستهدف مع ضرورة العمل علي عرض نتيجة الطلاب عليه بشكل واضـــح دون تدليل أو إحباط ، وهذا قد يتطلب الحصول علي اسم الطالب قبل البــــدء في الاختبار.



شكل رقم (٣٦)

نموذج لشاشة قائمة رئيسية
تتطلب من المتعلم ادخال اسمه



شكل رقم (٣٧)

نموذج لشاشة قائمة توضح
للتعلم أداؤه فيما تم دراسته
من موضوعات قبل البدء في العمل



شكل رقم (٣٨)

نموذج لشاشة تقويم
(سؤال موقوت)

لاحظ وجود زمن محدد للإجابة (٢٠) ثانية ، حيث أن هـــذا النمــوذج لشاشة اختبار موقوت ، وهذا يتطلب من المتعلم ضرورة الإجابة في خـــلال

وقت محدد وهو (٢٠) ثانية ، وإلا ظهر ما يفيد انتهاء الوقت دون مشاركة المتعلم في الإجابة .



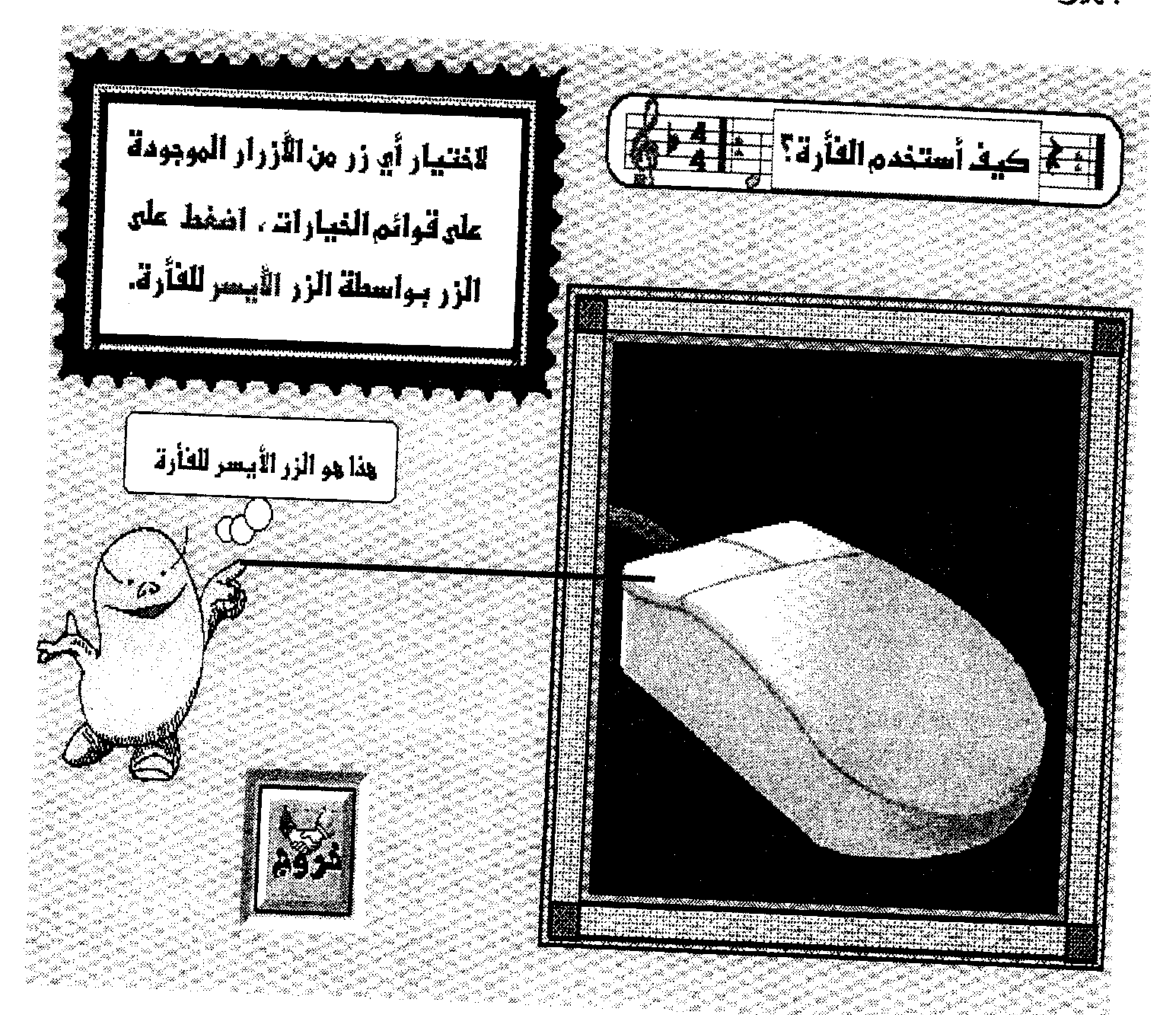
شكل رقم (٣٩)

نموذج لشاشة تقويم
(سؤال موقوت)

نموذج آخر لشاشة اختبار موقوت ، حيث يتطلب من المتعلم ضرورة الإجابة في خلال وقت محدد وهو (٢٠) ثانية .

٨- **شاشة المساعدة** : وهي نوعين من الشاشات :

- شاشة مساعدة من شانها أن تعين التلميذ علي الاستخدام الصحيح من البرمجية ، والتي عادة ما تضمن بعض الموضوعات الخاصة باستخدام وحدات الإدخال والإخراج للحاسوب : كاستخدام لوحة المفاتيح أو الفأرة أو تجهيز الطابعة للطباعة .



شكل رقم (٤٠)

نموذج لشاشة مساعدة تعين
المتعلم علي الاستخدام الصحيح للماوس

- شاشة مساعدة من شأنها أن تعين التلميذ علي فهم بعض العروض أو الأمثلة أو التمارين أو التدريبات ، حالة إخفاق التلميذ في ذلك.

٩- **شاشة تغذية راجعة :** والتي تتكون عادة من شاشة واحـــدة ، وتحتـــوى علي التغذية الراجعة المطلوبة للإستجابات الصحيحة والخاطئـــة علـــي حـــد سواء، والتي يمكن أن تكون نصا مقروء أو مسموعا أو صورة ثابتة معـــبرة أو متحركة أو لقطة فيديو.

١٠- **شاشة النهاية أو الخاتمة :** والتي تتكون عادة من شاشـــة واحـــدة أو أكثر، وهى التي عادة ما تودع المتعلم ببعض الكلمات الرقيقة التـــي تحفـــزه على معاودة العمل والتعلم من خلال البرمجية مرة أخـــري ، أو تتيـــح لـــه الفرصة للعودة مرة أخري للعمل بالبرمجية ، إذا ما كان قد اختار الخـــروج من البرمجية بطريقة عفوية وبدون قصد ، وقد تتيح لـــه أن يقضـــي بعـــض الوقت مع لعبة بسيطة هادفة لها علاقة بموضوع البرمجية.



شكل رقم (٤٣)

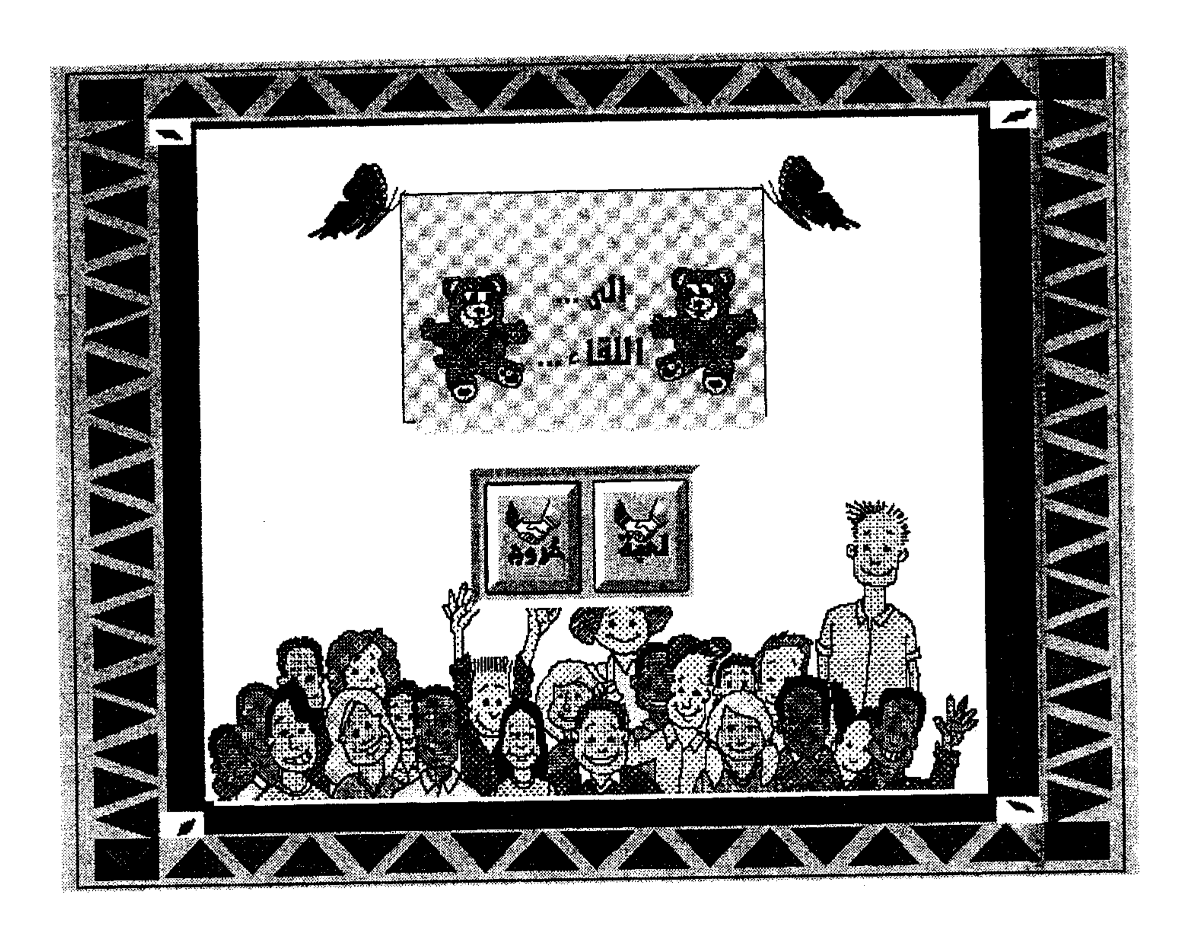
نموذج لشاشة خاتمة



شكل رقم (٤٤)

نموذج لشاشة خاتمة

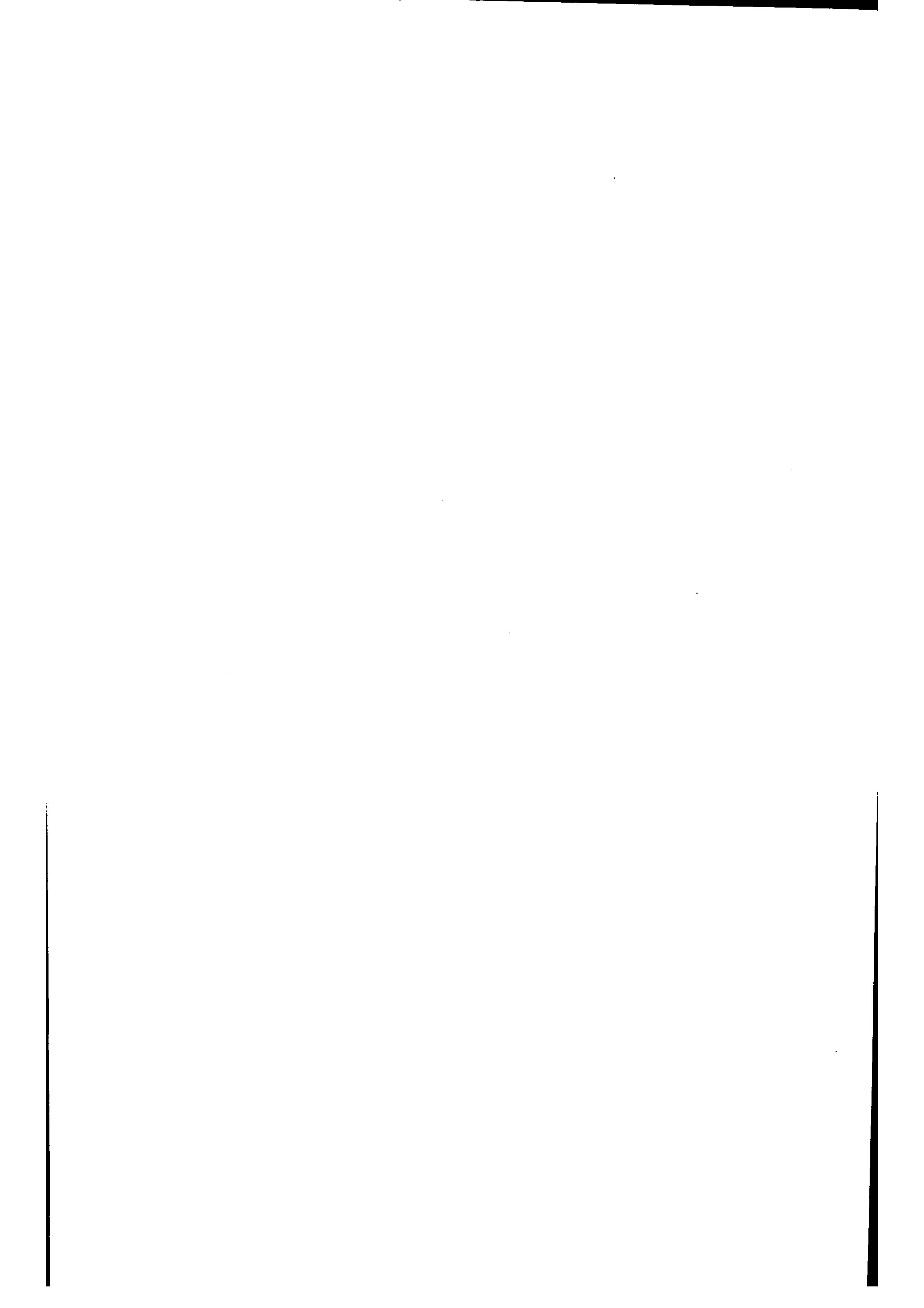
تأتي هذه الشاشة بعد استجابة المتعلم بـــ **نعم** فـــي الشاشـــة الخاتمـــة السابقة وهي الشاشة رقم (٤٣).

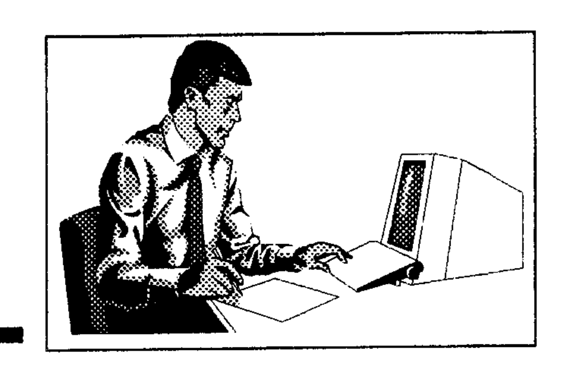


شكل رقم (٤٥)

نموذج لشاشة خاتمة

تأتي هذه الشاشة بعد استجابة المتعلم بـــ **نعم** في الشاشة الخاتمة السابقة وهي الشاشة رقم (٤٤).





الفصل الثالث

مرحلة

تنفيذ وإنتاج البرمجية



تمهيد

ينبغي أن تكون لدى المعلم الذي يقوم بتنفيذ البرمجية وبرمجتها خبرة باللغة المقترحة لبرمجة البرمجية ، وإمكانات استخدام الحاسوب تفوق ما لدى كل من المصمم ومعد السيناريو. هذا وينبغي أن لا يقدم المنفذ علي اتخاذ قرارات تتعلق بإضافة معلومات أو حذف بعض المعلومات أو التغيير في التسلسل أو التصحيح فيما هو معروض عليه من سيناريو قبل الرجوع إلى المصمم ومعد السيناريو. ينبغي علي منفذ البرمجية القيام بالإطلاع الشامل على محتوى الدرس أو محتوى البرمجية من البداية إلى النهاية ؛ حتى تتكون لديه الصورة الشاملة عن مجريات الأمور ، وتسلسل الأحداث وما سيستخدم أكثر من مرة وما سيستخدم مرة واحدة . هذا وينبغي علي المنفذ القيام بالمهام التالية ، قبل البدء بالتنفيذ :

١- التعرف علي إمكانات الحاسوب والتدرب علي تشغيله. وكذا الإطلاع علي مكونات معمل الحاسوب من أجهزة ومكتبة الصور *Clip Art Library* ومكتبة الأصوات *Sound (Mid & Wave) Library* ومكتبة لقطات الفيديو *Video Clip Library* .

٢- استخدام الحاسوب في استعراض بعض البرمجيات الخاصة بتعليم الرياضيات ونقدها بهدف الوقوف علي ما تتضمنه من نواحي القوة والضعف طبقا لقائمة بنود محددة ، ومناقشة كيفية تطويرها.

٣- التدرب علي استخدام الحاسوب في سماع العديد من المؤثرات الصوتية ، ومشاهدة العديد من الصور الثابتة والمتحركة ، والرسوم التوضيحية ، ولقطات الفيديو ، وكذا التدرب علي التحكم فيها ونسخها ، ودمجها ؛ وذلك بهدف اختيار ما يلزم لإنتاج البرمجيات المستهدفة.

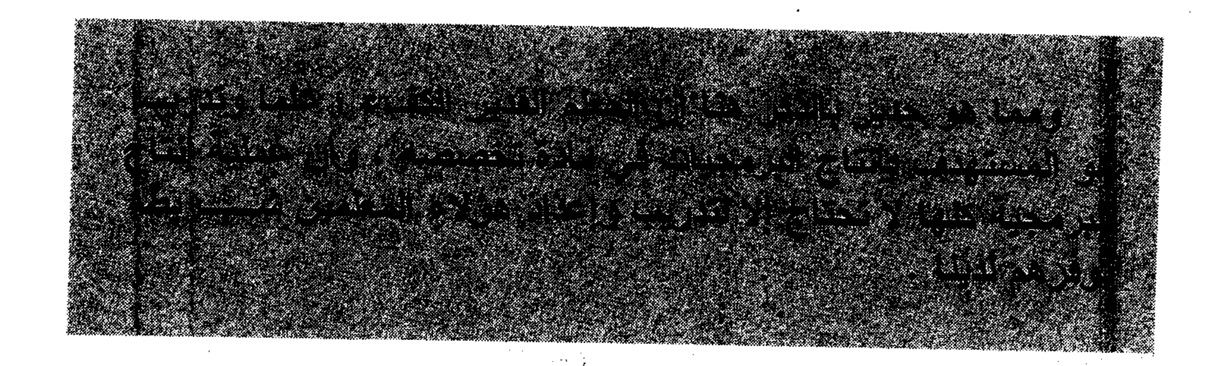
٤- التدرب علي استخدام الحاسوب في تسجيل المؤثرات الصوتية ، ورسم الصور الثابتة ، وإنتاج الصور المتحركة ، والرسوم التوضيحية ولقطات الفيديو ، بهدف تخليق ما يلزم إنتاج البرمجيات المستهدفة حالة عدم

توافرها بالمعمل وذلك باستخدام برمجيات خاصة بذلك مثل : *Sound Pro* ، *Corell Drow* ، *Adobe Primer* .

ومن الاختيارات المفضلة والمنشودة أن يكون معد سيناريو البرمجية هو نفس المعلم الذي سيقوم بتنفيذ البرمجية وبرمجتها على الحاسوب ؛ وهو نفسه الذي قام بتصميم البرمجية ، وتجميع ما يلزم إنتاجها من مواد علمية وتعليمية وأنشطة ومفردات اختبار ، ... الخ.

بعد الانتهاء من تنفيذ البرمجية بالكامل يقوم المنفذ بتجريب البرمجية ككل؛ وذلك عن طريق تشغيلها ورؤيتها من وجهة نظر التلميذ لاكتشاف أي أخطاء ، أو تعديلات ينبغي أن يقوم بها، وبعد الانتهاء من كتابة التدريبات فإنه يقوم بتشغيل البرمجية لرؤية هذا الجزء الخاص بالتدريبات وفي هذه الحالة يقوم المنفذ بالإجابة الصحيحة على جميع الأسئلة وتسجيل أي ملاحظات تتعلق بهذا المسار، ثم يقوم بتنفيذ البرمجية مرة ثانية ، متخذا مسارا مختلفا ؛ مثل الاستجابة خطأ عن كل سؤال في أول محاولة وفي المحاولة الثانية يستجيب بطريقة صحيحة ، ومسارا آخر مثل الاستجابة خطأ في المحاولة الأولى والثانية . والهدف من اختبار المسارات المختلفة للبرمجية هو التأكد من خلو البرمجية من أي أخطاء أو سوء تقدير.

بعد الانتهاء من مراجعة البرمجية من قبل المنفذ ؛ ينبغي أن تراجع مرة ثانية من قبل معد السيناريو ، وكذلك مصممها ؛ للتأكد من أن البرمجية تعمل بالطريقة التي حددها وبالشكل الذي يرغبان فيه . وفي بعض الأحيان قد تجرى بعض التعديلات التي لم تكن موجودة أصلا في التصميم أو في السيناريو ، ويكون سبب ذلك هو أن الإخراج النهائي للبرمجية قد كشف عن أشياء لم تكن في الحسبان أثناء التصميم وإعداد السيناريو ، وهكذا يستمر التفاعل بين المنفذ أو المبرمج من ناحية ، ومعد السيناريو ومصممها من ناحية أخرى ، إلى أن تصل البرمجية في حالتها النهائية إلى شكل مستقر ومقبول.



مكتبات لإنتاج الأوساط المتعددة

يحتاج منفذ برمجيات الوسائط المتعددة إلى العديد من وسائط المالتيميديا والمتمثلة في الصور الثابتة والمتحركة والأصوات ولقطات الفيديو ، وعليه أن يبحث ويتجول داخل مكتبات الأوساط المتعددة لعله يعصر علي مبتغاه – وهي مكتبات تتضمن علي العديد من العناصر المعدة بشكل مسبق وبشيء من الإتقان– وفي أغلب الأحيان يحصل المنفذ علي اكثر من ٨٠% مما يبحث عنه داخل تلك المكتبات ، وعليه أن ينتج الأوساط التي لم يعصر عليها بمكتبات الوسائط المتعددة المتوافرة لديه . ومن هذه المكتبات :

مجموعة	*Sonic Waves 3000*	مجموعة	*PhotoSphere Library*
مجموعة	*Terra Incognita*	مجموعة	*3000 Photo Gallery #2*
مجموعة	*The Object Series*	مجموعة	*The Publishers Depot*
مجموعة	*MGDigitalAtlas*	مجموعة	*DigiClips Special Edition*
مجموعة	*KPT Power Photos*	مجموعة	*CD-ROM transportation Set*
مجموعة	*CyberManager 95*	مجموعة	*Firstlight VideoClips*
مجموعة	*Media Commander*		

تعتبر لغة فيجوال بيسك المدعمة بالرسومات والصور والتي تعمل ضمن بيئة ويندوز *Windows* من ضمن الأنظمة الخاصة لتأليف البرمجيات التعليمية، حيث تقدم العديد من التسهيلات التي تمكن المستخدم من إنتاج الدروس التعليمية كما تتيح له إمكانية الاستفادة من إمكانات بيئة ويمب

WIMP المتوفرة في ويندوز إضافة إلى الخطوط العربية . ورغم مزاياها المتعددة عن لغات البرمجة الأخرى لم تعفى المستخدم تماما من مهمة البرمجة . على المستخدم أن يعد البرمجية التعليمية في صورة تطبيق *Application* والذي يتكون عادة من مجموعة نوافذ *Forms* وموديولات *Modules* وبناءات منطقية *Codes* تعمل معا كبرنامج واحد. وعليه فقد يحتاج المستخدم إلى خبره سابقة في البرمجة ليتمكن من بناء البناءات المنطقية (البرنامج) بلغة فيجوال بيسك والتي تختلف في بعض المواقف كبيئة عن بقية إصدارات البيسك الأخرى (*QBASIC* ، *BASICA* ، *GWBASIC*،...) ، والشبيهة إلى حد ما بلغة بسكال.

تتكون الشاشة الرئيسية في نظام فيجوال بيسك *Visual BASIC* (VB) **الإصدار الثالث** من خمس نوافذ رئيسية ؛ قد تظهر متراكبة فوق بعضها البعض عند بدء التشغيل إلا أنها قابلة للتحريك من مواقعها وتعديل حجمها كما هو الحال في تطبيقات النوافذ *Windows*

لقد بدأت لغة فيجوال بيسك عام ١٩٩١ كوسيلة لتصميم البرامج التي تعمل على غرار ويندوز ، وبدأت تلك اللغة بالإصدار الأول (1.0) ، ثم الإصدار الثاني (ver 2.0) ، ثم ظهر الإصدار الثالث (ver3.0) في منتصف عام ١٩٩٣ ، وقد جاء في صورة حزمتين : الحزمة القياسية *Standard* ، وحزمة المحترفين *Professionals* والتي تحتوى بالطبع على كثير من الإضافات التي تساعد المبرمج على إنجاز الكثير من الأعمال بسهولة ويسر ، وتوجد الآن الإصدار الرابع (ver 4.0) والذي يدعم أنظمة (٣٢ بت) ، ويوجد الآن الإصدار الخامس والسادس.

ومما هو جدير بالذكر أن هناك تشابه ما بين مفردات لغة فيجوال بيسك وإصدارات لغة بيسك التقليدية ، كما أن هناك اختلافات جوهرية بينهما في مفهوم البرمجة نفسه : فالمفهوم التقليدي للبرمجة يعتمد على الخطوات المتتابعة التي تمثل منطق حل المشكلة بينما المفهوم الذي تتبعه لغة فيجوال بيسك هو البرمجة الموجهة بالأحداث ***Event - driven Programming*** : لذا تنتمي لغة فيجوال بيسك إلى ما يسمى بلغات البرمجة الموجهة نحو الأهداف ***Object-Oriented Programming*** وهى لغات جديدة ظهرت بظهور بيئة النوافذ ويندوز مثل لغة *Visual C++* ، ولغة *Visual Pascal* حيث أصبحت

البرمجة نحو الأهداف اتجاها جديدا في معالجة بيانات الأهداف *Object Data* وكود البرمجة *Programming Code* في نفس الوقت ؛ وبذلك فان الهدف يصبح كائنا مستقلا بذاته بما يحتويه من بيانات تمثل هويته ، وكود يمثل سلوكه : إن هذا المفهوم الجديد في البرمجة جعل عملية البرمجة اكثر سهولة ولاسيما في بيئة ويندوز.

والتطبيق الموجه للأحداث يستجيب للأحداث التي تحدث في بيئة الحاسوب : مثل الضغط على أحد أزرة الفأرة أو لوحة المفاتيح . وقد يكون مصدر الحدث هو المستخدم أو الحاسوب أو البرنامج نفسه أو برنامج آخر يعمل في نفس الوقت على اتصال بالبرنامج الحالي . وعندما يأخذ الحدث طريقه إلى التنفيذ ، يبدأ الروتين المناسب في العمل ، فعلى سبيل المثال : عندما يضغط المستخدم على زر معين بالشاشة فان البرنامج التطبيقي يبدأ بالتعامل مع المستخدم بان يقدم معلومة بالشرح والتحليل ؛ إن يتناول حل مشكلة أو يقدم أمثلة محلولة ، أو يطلب من المستخدم تأدية اختبار معين ، وقد يفتح نافذة جديدة تحتوى على بعض المعلومات ويطلب من المستخدم إدخال بعض البيانات في الموقع (الصناديق) المخصصة لذلك.

والمبرمج لا يبدأ من الصفر ، فالنوافذ جاهزة والأزرة جاهزة ، وما عليه إلا تنظيمها بالصورة المناسبة والمطلوبة وتوجيهها وفقا للأحداث المتوقعة ، إضافة إلى أن لغة فيجوال بيسك تتضمن بعض الروتينات الجاهزة الخالية من الكود والتي يضع فيها المبرمج تعليماته الخاصة.

إن الاهتمام الأكبر في مجال البرمجة الموجهة بالأحداث ينصب على تصميم واجهة المستخدم ***User Interface*** والتي تعرف بما يسمى الوصلة البينية ، والتي تتكون عادة من النوافذ وماتضمنة تلك النوافذ من أدوات تحكم : **أي أن الوصلة البينية للمستخدم هي شاشة**

....

(*) يحلو للبعض أن يطلق على لغات البرمجة الموجهة نحو الأهداف السميات التالية:

Object-Oriented Programming (OOP)

Object-Oriented Programming System (OOPS)

Object-Oriented Programming LAnguage (OOPLA)

البرنامج المخصصة للتعامل مع المستخدم ، والتي قد تحتوى على قائمة اختيارات أو على نافذة ذات أزرار وأيقونات ، وهى في النهاية تصل ما بين المستخدم وما بين كود التطبيق. وان عملية تصميم وإنشاء تلك الوصلات البينية (واجهات المستخدم) كما عرفنا فيما سبق ، لا تحتاج إلى كتابة أية تعليمات أو عبارات فهي عملية فنية بحتة تشبه التصميم بالنماذج الجاهزة . أما الكود الذي يكتبه المبرمج فهو عبارة عن مجموعة من الروتينات المنفصلة التي ترتبط بأدوات التحكم المختلفة وتصف كيفية استجاباتها للأحداث.

هذا وتحتوى بيئة لغة بيسك المرئية على أهداف أساسية *Object* هي النماذج *Forms* وأدوات التحكم *Control Units* ، ويمثل النموذج النافذة التي نراها على شاشة الحاسوب أثناء تشغيل البرمجية ، أما أدوات التحكم فهي حواس النافذة التي تتلقى عن طريقها الأوامر مثل الأزره ، وصناديق النصوص ، وصناديق الحوار، ... .

أي أن مشروع البرمجية يتكون من مجموعة من النماذج أو الشاشات ووحدات تحكم تستجيب لمجموعة من الأحداث التي من المتوقع أن يؤديها المتعلم : كأن يضغط على مفتاح ما ، أو يفتح نافذة ، أو يكتب في حقل ، أو يحرك مؤشر الفأرة على هدف ما (وحدة تحكم أو نافذة) وينقر زر الفأرة .

هذا ويرتبط كل هدف من الأهداف في لغة بيسك المرئية بمجموعة من الأحداث *Procedures* ، حيث تستطيع أن تختار من هذه الأحداث المجموعة المناسبة التي تجعل أهدافك تستجيب لها بالطريقة التي تحددها أنت من خلال الأكواد المكتوبة.

الفصل الرابع

مرحلة
تجريب وتطوير البرمجية

تمهيد

ينبغي أن تعرض البرمجية التي يتم إعدادها – مرحلة التجريب والتطوير *Development* – علي عدد من الموجهين والمعلمين ، وكذا خبراء المناهج وطرق التدريس ، وأساتذة علم النفس التربوي ، إضافة إلى عرضها عمليا علي عينة من التلاميذ مأخوذة من المجتمع الأصلي الذي ستطبق فيه هذه البرمجية ، وعلى ضوء مقترحات الموجهين والمعلمين وخبراء المناهج وطرق التدريس ، وكذا أساتذة علم النفس التربوي ؛ وذلك من خلال قوائم التقويم المعدة لهذا الغرض ، وفي ضوء مواقف عينة التلاميذ والطلاب ؛ يجري مزيدا من التعديلات على البرمجية إذا لزم الأمر ، إلى أن تصل إلى مستوى يسمح بنشرها وتعميمها على نطاق واسع .

ينبغي أن نبدأ من الآن ، ودون إبطاء في إعداد كافة المعلمين وتدريبهم على إنتاج ما يقومون بتدريسه من مقررات في صورة برمجيات تعليمية .

مع الزيادة الكبيرة في عدد البرمجيات التعليمية المقدمة عن طريق الحواسيب ، فإنه يصبح **من الضروري والهام جدا أن تتوفر لدينا بعض المعايير الخاصة بتقييم مثل هذه البرمجيات** . أضف إلى ذلك أن هدفنا هو إعداد برمجيات تعليمية ذات جودة عالية ؛ تتطلب الكثير من الخبرات ، ولذا فإن أغلبية المعلمين سوف يعتمدون ، ولفترة ، على الكثير من البرمجيات التي يعدها الآخرون، وفي بعض الأحيان فإن مثل هذه البرمجيات التعليمية يعدها أفراد غير تربويين تتوفر لديهم خبرات كبيرة عن البرمجة ، ولا يتوفر لديهم إلا القليل عن الكيفية التي يتعلم بها الأفراد. لذلك نرى الشكوى قد كثرت فيما يتعلق بجودة البرمجيات التعليمية ، لعدم توفر الكتيبات

المصاحبة أو الكاتالوجات أو أدلة التشغيل ، عدم المرونة، واحتوائها على مصطلحات فنية غير ملائمة ، مع عدم العناية بدقة الاستجابات الخاطئة.

إننا نشجع اشتراك المعلمين في عملية إعداد البرمجيات التعليمية للحواسيب ، إن لم يكن على مستوى التنفيذ فعلى الأقل على مستوى التصميم والإعداد والتجهيز ، وإعداد السيناريو أو حتى على مستوى التقويم . وهذا لا يعنى أننا نقترح أن يكونوا خبراء في البرمجة، ولكن بالقطع سيكون **المعلمون قادرين على القيام بتلك الوظائف الجديدة بشكل أفضل إذا أحسن إعدادهم وتدريبهم وإعادة تأهيلهم ، وهذا يعني البدء في ذلك فورا دون إبطاء ، حتى يمكن الوصول إلى النتائج المرغوب فيها بالنسبة للمتعلم.**

إن مصطلح التقييم يتضمن المعايير والمستويات. فتقويم أو تقييم برمجية تعليمية يتضمن معرفة خصائص البرمجية الجيدة، وتحديد خصائص هذه البرمجية ليس بالأمر السهل؛ ففي العملية التعليمية مدى متسع لنواتج التعلم المقبولة على وجه العموم، وكذلك بالنسبة للاستراتيجيات التي يختار منها. **ورغم أنه لا يمكننا تحديد خصائص محددة ينبغي أن تشتمل عليها كل برمجية تعليمية جيدة. فإنه يمكننا أن نقترح بعض المبادئ والأسس العامة التي يمكن تطبيقها، والتي تجعل عملية التقويم – أو التقييم – لمنتج تعليمي أكثر موضوعية.**

الخصائص العامة للبرمجية التعليمية الجيدة

إن الغرض الأساسي لعملية التدريس هو تسهيل عملية التعلم على وجه العموم ، ولقد قام كل من **جانيه** *Gagne* **وبريجز** *Briggs* **وبرونر** *Proner* **وبلوم** *Bloom* **وبياجيه** *Paiget* بتحديد بعض المواقف التدريسية التي تصف الكيفية التي يتم بها إنجاز عملية التدريس ، هذه المكونات أو الأحداث التدريسية يمكن أن تمدنا بالهيكل الأساسي لتصنيف خصائص البرمجيات التعليمية ويمكن عرضها على النحو التالي:

١- تشد الانتباه.
٢- تبلغ المتعلم الهدف.
٣- تثير وتساعد على تذكر المتطلبات السابقة للتعلم.
٤- تقدم مواد تعليمية مثيرة.
٥- ترشد المتعلم.
٦- تقود إلى الإنجاز.
٧- توفر تغذية راجعة تتعلق بتصحيح الإنجاز.
٨- تقوم الإنجاز.
٩- تساعد على التذكر ونقل أثر التعلم.

وليس من الضروري أن تتوفر كل هذه المهام في كل برمجية تعليمية؛ فأحيانا قد يطلب من الطلاب أن يتحملوا جزءا من شروط تعلمهم بمجهوداتهم الخاصة، وفى الغالب فإن أحداثا معينة تكون متوفرة في الموقف التعليمي ككل، وخارج البرمجية التعليمية موضع التقويم . فقد تكون هناك برمجية تستخدم كجزء من برمجية تعليمية أكبر. وقد يكون هدفها المحدد هو إثارة الدافعية فقط أو التدريب أو الاختبار.

وفى الموقف التعليمي ينبغي أن نأخذ بعين الاعتبار بعض الخصائص المحددة للمتعلمين ومحددات أفضل بيئة تعليمية : كدرجة الإضاءة، الوقت المخصص للتعلم خلال اليوم ... الخ ؛ حيث تختلف من فرد لآخر ، كما أن الجهاز الحسي المستقبل الأكثر فاعلية لمشاركة الأفراد في التعلم بالعمل : كالنظر ، السمع ، أو اللمس ، يختلف باختلاف الأفراد . وأن درجة الاهتمام بموضوع معين أو بالطريقة المستخدمة في عرض هذا الموضوع ليست واحدة بالقطع ، بالنسبة لكل طالب. إن تبنى استراتيجية للتدريس للاستفادة ، إلى أقصى حد ، بخصائص المتعلم تعد صعبة ، نسبيا ، في التدريس داخل الفصل التقليدي ، وبالمقارنة فإن تكنولوجيا التدريس بواسطة الحاسوب لديها قابلية وقدرة أكثر على تبنى حاجات المتعلم الفرد.

خصائص الموقف التعليمي الجيد

ينظر للموقف التعليمي هنا على أنه يتكون من المجموع الكلى للعوامل التي تساعد على تعلم وحدة تعليمية معينة. وهذه العوامل الأساسية تكاد تكون هي نفسها ؛ بغض النظر عن الأساليب المستخدمة في التدريس ، تعد مسئولة عن العبء التدريسي، فإذا استخدم أسلوب المحاضرة أو برمجية من خلال الحاسوب أو رحلة مستقلة لمتحف أو أي نشاط آخر، فإن المواقف التعليمية الجيدة ينبغي أن تتمتع بخمس خصائص هي : **التعلم ينبغي أن يكون صحيحا، والمتعلم ينبغي أن يكون مستعدا، والحاجات التعليمية ينبغي مراعاتها أو تسهيلها، والاستيعاب ينبغي أن يكون عمليا. والتعلم ينبغي أن يكون فعالا،** وسوف نتعرض لكل من هذه الخصائص بشيء من الاختصار.

١- **التعلم ينبغي أن يكون صحيحا :** قد يكون هناك نشاط تدريسي معين يؤدى إلى التعلم بشكل جيد ، ولكن ما لم يكن هذا الجزء في اتساق مع الفلسفة التي وراء النظام التعليمي ، ويساعد على تحقيق الأهداف العامة للمقرر موضع الدراسة ، الذي هو جزء منه ، فإن هذا النشاط التدريسي لا قيمة له . وهذا المعيار ينبغي مراعاته ؛ حيث إن الأهمية وراء فحص البرمجية إضافة إلى التأكد من دقتها ، هو البحث عن هذا الهدف.

٢- **المتعلم ينبغي أن يكون مستعدا :** إن النجاح في التعلم من المحتمل أن يعتمد على استعداد المتعلم ، أكثر من اعتماده على المواد التعليمية المستخدمة. وإن الاستعداد له علاقة بالعقل والانفعالات والظروف الفيزيائية. فالاستعداد العقلي للمتعلم يعد الأكثر وضوحا وأهمية : فالمهارات العقلية والاستراتيجيات المعرفية تحدد الاستعداد العقلي. أما الاستعداد الانفعالي أو العاطفي سيشتمل على مفهوم التعزيز الوارد في نظريات التعلم. وإن تحقيق درجة عالية من حب الاستطلاع المعرفي لدى المتعلم، مع الاحتفاظ بالحد الأدنى من مستويات القلق يعد من العوامل الهامة. ومساعدة الطالب على تنمية شعور قيمة الذات ومفهوم الدافعية تعد كذلك عناصر في الاستعداد العاطفي للمتعلم وعوامل الاستعداد الفيزيائي تشتمل على التحرر من الحاجات

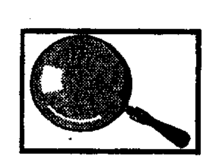
الفيزيائية والجسدية، والتيقظ، ونظام عصبي سليم وصحي، وفى الغالب القوة والتحمل، إن المواد التعليمية وحتى البيئة التعليمية الحالية من النادر أن يكون لها تأثير على الاستعداد الفيزيائي للمتعلم. وبالتأكيد فإن المتعلم غير المعد فيزيائيا لا يتوقع له النجاح.

٣- **الحاجات التعليمية ينبغي مراعاتها أو تسهيلها** : تتعامل هذه الخاصية والخاصيتان التاليتان مع الاستيعاب والفاعلية ، وتعد هذه الخصائص الأكثر وضوحا، ولها صلة مباشرة بكاتب سيناريو البرمجية التعليمية أو بمصممها . ولقد اقترح عديد من تربويون إجراءات محددة تميز جودة البرمجيات التعليمية المقدمة عن طريق الحاسوب . وإلى حد ما، فإن عناصر إدارة التعلم أو المهام التدريسية التي سبق لنا تحديدها، تعد متضمنة في بعض الخصائص الأخرى للبرمجية التعليمية الجيدة . ورغم أن التعلم بحده الأدنى يمكن أن يحدث دون كل هذه العناصر ، فإن توفرها يدل على طراز تدريسي منظم يعد مرغوبا فيه ومستهدفا على وجه العموم.

٤- **الاستيعاب ينبغي أن يكون عمليا** : جعل الاستيعاب عمليا يعنى جعل التعلم في متناول الطالب. وأحد أهم العناصر الواضحة والضرورية للبرمجية التعليمية الجيدة هو مدى توفر المعلومات . إن تنمية المهارة والاتجاهات والتذكر البسيط للمعلومات ينبغي أن يعتمد على كل من المعلومات المتاحة وما يعرفه المتعلم بالفعل . كما ينبغي تقديم المعلومات في ترتيب منطقي ؛ وإن مفهوم التسلسل المتصل والمستمر وتوفيره ينبغي أن يتلاءم مع منطق عرض المعلومات . إن أهمية التكرار أمر مسلم به ، وتعد إجراءات التحكم أو مهارات تعلم الطالب مثل : تسهيل عملية تسجيل الملاحظات ، من الأمور الهامة في مساعدة الذاكرة ذات المدى القصير. إضافة إلى أن توفير المعارف ذات الصلة مع توفير وقت كاف يعد من الأمور التي نحتاجها في ترميز المعلومات التي ستخزن في الذاكرة ذات المدى الطويل ، وفى فك ترميز تلك المعلومات وقت استدعائها. وينبغي مراعاة عدم تحميل ذاكرة المدى القصير أكثر مما ينبغي.

٥- **التعلم ينبغي أن يكون فعالا** : البرمجية التعليمية ربما تكون ناجحة من حيث تحقيق أهدافها . ولكن إذا لم تكن هذه البرمجية فعالة ، فإنه ليس من المحتمل اعتبارها ذات قيمة عظيمة ، مع الأخذ في الاعتبار أن خصائص

المتعلمين تعد بكل وضوح هامة في زيادة فاعلية التعلم . ولهذا صلة بالفكرة القائلة : إن وقت الطالب يجب استخدامه بتحفظ . وكذلك فإن التكلفة المالية ينبغي ألا تزيد عن تكاليف البرمجيات الأخرى المشابهة . كما ينبغي استخدام وقت المعلم وطاقاته بفاعلية أكثر.



معايير تقييم البرمجيات التعليمية

عند مراجعة وتقييم البرمجيات التعليمية – جاهزة الإعداد – والتي تعرف أحيانا بالمقررات المبرمجة *Courseware* بهدف التوصية بشرائها واستخدامها في مدارسنا أو في منازلنا فإنه من الضروري أن نحتفظ ببعض البيانات بصفة دائمة ومنظمة بحيث يمكن الرجوع إليها في أي وقت نشاء ، كما ينبغي أن تتوفر نماذج مختلفة لتقييم الأنماط المختلفة للبرمجيات التعليمية ، وهنا سوف نقدم نموذجا – قائمة بنود – يمكن الاسترشاد به عند تقييم تلك البرمجيات ، علما بأن عملية تقييم البرمجيات التعليمية تعد ذاتية ونسبية إلى حد ما ؛ فما يراه أحد الأفراد عملا رائعا ، يراه شخص آخر شيئا عاديا أو غير ذي أهمية ، لذلك ينبغي عدم الاعتماد على نتائج تقييم فرد واحد، كما يفضل أن يتم تجريب هذه البرمجيات في مواقف فعلية يستخدمها التلاميذ بطريقة تمكن القائمين على ملاحظتهم من تقييمها بصورة فعالة. ويجب ألا يهمل القائمون بالتقويم رد فعل المعلمين تجاه استخدام تلك البرمجيات.

في بداية القائمة ، خصص جزء لجمع المعلومات الخاصة بالتعريف بالبرمجية مثل : **وصف البرمجية** ، **متطلبات التشغيل** ، **وسياسة النسخ الاحتياطية** ، ثم اشتملت القائمة على أربعة محاور رئيسية هي : **خصائص المحتوى** ، **خصائص استخدام الطالب** ، **خصائص استخدام المعلم** ، **خصائص تشغيل البرمجية**. صممت مفردات القائمة بمدرج ليكارت المكون من أربعة اختيارات توضح درجة توفر الخاصية ، وهى : **ممتازة (٣)** ، **جيدة (٢)**، **ضعيفة (١)** ، **لا تنطبق (٠)**. وعلى القائم بعملية التقييم ؛ بعد الانتهاء من عملية التقدير طبقا لبنود القائمة أن يقوم بجمع درجات كل معيار أو محك ، وحساب النسبة المئوية لكل معيار منسوبة إلي الحد الأقصى للنقاط الممكنة لهذا المعيار ، وبالتالي حساب درجات والنسبة المئوية لبنود القائمة ككل.

المعلومات الخاصة بالتعريف بالبرمجية

يمكن للقائمين بالتقويم ، معرفة المعلومات الخاصة بالتعريف بالبرمجية عن طريق التجريب أو من خلال الإطلاع على دليل التشغيل المرفق مع البرمجية أو من خلالهما معا.

• وصف البرمجية

- **اسم البرمجية** من المفيد أن يسجل المقيمون اسم البرمجية .
- **الناشر** فهناك من يتمتع من الناشرين بسمعة جيدة على خلاف غيرهم.
- **سنة النشر** لضمان الحداثة والتطوير.
- **النمط** ويقصد به تحديد نمط البرمجية.
- **المادة المبرمجة** ويقصد بها مجال البرمجية ؛ قراءة - حساب - فيزياء ...الخ.
- **الموضوع** ويقصد به الموضوع أو رؤوس الموضوعات التي تعرضها البرمجية.
- **عدد الدروس** ويقصد به عدد الدروس التي تحتويها البرمجية.
- **المستوى الصفي** ويقصد به المستوى الدراسي أو الصفوف التي يمكن لطلابها الاستفادة من محتوى البرمجية.
- **بيئة التشغيل** هل تعمل البرمجية في بيئة الدوس *Dos* أو في بيئة الوندوز *Windows*
- إمكانية التعامل مع شبكة انترنيت هل تتعامل البرمجية مع أحد الشبكات المحلية أم تتعامل مع شبكة انترنيت.

• متطلبات التشغيل

• **نوع الحاسوب المطلوب** : ينبغي تحديد ماركة وطراز الحاسوب الذي أعدت له البرمجية ، فقد لا تصلح البرمجية لبعض الطرازات.

• **سعة الذاكرة المطلوبة** : ينبغي أن يتوفر للحاسوب سعة ذاكرة معينة كي يمكن تشغيل برمجيات محددة عليه ، وبالتالي فإنه من المفيد معرفة الذاكرة المطلوبة لتشغيل البرمجية المستهدفة .

• **بطاقة الإظهار المطلوبة** : في كثير من الأحيان تكون هناك برمجيات ملونة ، حيث يعد اللون أساسيا لاستثارة دافعية الطلاب للإنجاز ، وهذا يتطلب بطاقات خاصة للإظهار مثل : VGA ، EGA ، SEGA.

• **متطلبات إضافية** : هناك برمجيات معدة للعمل مع الحواسيب بشرط أن يتوفر لها بعض مكونات مادية إضافية *Hardware* كالطابعة *Printer* مشغل أقراص معين *Disk Drive* ، بطاقة خاصة للصوت *Sound Card* ... الخ.

• سياسة النسخ الاحتياطية

تعد سياسة النسخ الاحتياطية *Backup Policy* ، من حيث مبدأ قبولها من عدمه ، هامة لقرار الشراء ، وذلك لارتباطها بالتكلفة الكلية للبرمجية ، وعدد المستخدمين ، وفيما يلي بعض الأمثلة لهذه السياسات :

١ - برمجيات تأتى بالأقراص الرئيسية مشتملة على نسخة احتياطية لها، وتعد مثل هذه السياسة مقبولة عادة ، وقد يكون من الترتيبات المتاحة أن يسمح بإرسال القرص الرئيسي أو الاحتياطي لمكان الشراء لإحلاله بآخر مقابل مبلغ رمزي.

٢ - برمجيات تأتى فقط بالأقراص الرئيسية ، وبدون الأقراص الاحتياطية ، على أن يتم الحصول على بديل لها حالة تلفها أو حدوث أعطال لها تمنع تشغيلها بشكل جيد ، وفي هذه الحالة لابد من إرسالها إلى مكان الشراء للاستبدال ، وعليه لابد من وضع ذلك في الاعتبار ؛ حيث يتطلب ذلك الاستغناء عن استخدام البرمجية فترة من الوقت.

٣ - برمجيات تأتى بالأقراص الرئيسية ، على أن يتم إرسال النسخ الاحتياطية بعد استيفاء بطاقة التسجيل - بطاقة ملكية البرمجية - وإرسالها

إلى مكان الشراء ، وتعتبر هذه السياسة مقبولة إلى حد ما ، وينبغي في هذه الحالة استيفاء بطاقة التسجيل وإرسالها فور وصول البرمجية.

٤ – برمجيات تأتى فقط بالأقراص الرئيسية ، ولا يوجد سياسة للإحلال أو لتوفر نسخ احتياطية ، وتعتبر هذه السياسة غير مقبولة ، إلا إذا كانت تكلفة شراء تلك البرمجيات رخيصة جدا وتسمح بتعدد شراء النسخ.

٥ – برمجيات غير ممنوعة من النسخ ، أي يمكننا عمل نسخ احتياطية وبأعداد غير محددة، وبالطبع تعتبر هذه السياسة من أفضل السياسات، وهذا يتطلب ضرورة قيامنا بعمل نسخة أو نسخ احتياطية بمجرد وصول النسخة الرئيسية وحفظها في مكان أمين.

معيار خصائص المحتوى

ينبغي أن يتوفر في البرمجية ، فيما يخص المحتوى التعليمي موضوع البرمجية ، الخصائص التالية :

١– تتبنى البرمجية نظريات تربوية صحيحة في عرضها للمحتوى.
٢– دقة المحتوى وسلامته العلمية.
٣– تستخدم البرمجية أنشطة تعليمية مقبولة
٤– تناسب مقدار التعلم مع ما يستغرقه المتعلمون من وقت.
٥– وضوح التسلسل والتتابع المنطقي للدروس.
٦– يراعى تحقق الأهداف المذكورة.
٧– الاستخدام الملائم للأصوات والألوان.
٨– إمكانية طبع أي جزء من المحتوى.
٩– الاستخدام الملائم للرسوم والنماذج المتحركة.
١٠– الترابط بين أسلوب التمثيل وحركة الرسوم والنماذج بأهداف المحتوى ومضمونه.

معيار خصائص استخدام الطالب

ينبغي أن يتوفر في البرمجية ، فيما يخص استخدام الطالب لها ، الخصائص التالية :

١- لا تتطلب معرفة مسبقة للطالب بالحاسوب.
٢- حث الطلاب على التعاون والعمل المشترك.
٣- لا تتطلب من الطالب الرجوع لدليل التشغيل.
٤- توفر للطالب ملخصا عن أدائه.
٥- تغذية راجعة فعالة للاستجابات الصحيحة والخاطئة على حد سواء.
٦- التغذية الراجعة الموجبة أكثر جاذبية من التغذية الراجعة السالبة.
٧- تتيح للطالب أن يتحكم في معدل عرض المعلومات.
٨- تتضمن وظائف لتحليل أخطاء الطلاب.
٩- تتيح للطالب أن يتحكم في تسلسل محتويات الدرس.
١٠- تتيح للطالب أن يتحكم في اختيار الدرس.
١١- تتيح للطالب أن يختار العودة لمراجعة أجزاء معينة من درس معين.
١٢- تتيح للطالب أن يختار أنماطا مختلفة للعرض.
١٣- تتضمن البرمجية عدة مستويات من الصعوبة والسهولة .
١٤- سهولة قراءة النصوص المعروضة على الشاشة باستخدام حروف ذات أحجام مناسبة.
١٥- تتضمن البرمجية وظائف مساعدة *ON- Line Help*
١٦- التقليل من الاعتماد على المعلم.

معيار خصائص استخدام المعلم

ينبغي أن يتوفر في البرمجية ، فيما يخص استخدام المعلم لها ، الخصائص التالية :

١- عرض الأهداف التعليمية بوضوح.
٢- تتكامل الأهداف مع المحتوى .
٣- تتيح للمعلم أن يتحكم في مستويات صعوبة بعض الصياغات.
٤- تتيح للمعلم أن يغير من قوائم المفردات كالكلمات والمسائل.
٥- توفر كتيبات للمعلم أو مواد تعليمية مساعدة.
٦- توضح دور المعلم.
٧- تقترح خططا للتدريس.
٨- توفر كراسات عمل مفيدة للطالب.
٩- توفر أنشطة إثرائية للطالب سريع التعلم.
١٠- توفر أنشطة علاجية للطالب بطيء التعلم.
١١- توفر ملخصا لأداء كل طلاب الفصل.
١٢- تقبل البرمجية وتقدم أجوبة متنوعة.
١٣- تقترح استخدام أنشطة ومصادر تعليمية أخرى.
١٤- إمكانية طبع النتائج المسجلة.
١٥- إمكانية توليد مفردات الاختبارات وطباعتها.

معيار خصائص تشغيل البرمجية

ينبغي أن يتوفر في البرمجية ، فيما يخص تشغيلها الخصائص التالية :

١- سهولة الدخول إلي البرمجية والخروج منها.
٢- وجود دليل استخدام البرمجية بصياغة واضحة.
٣- ترابط عرض دروس البرمجية على الشاشة مع المضمون.
٤- التنسيق على الشاشة واضح وجميل.
٥- تسمح للمستخدم بتصحيح أخطاء الكتابة.
٦- سهولة استخدام البرمجية .
٧- تتيح اختيار أجزاء محددة من محتوى البرمجية .
٨- نصوص البرمجية سليمة اللغة واضحة المعنى.
٩- تتيح البرمجية تشغيلا موثوقا ، وذلك بعدم تعطيلها حالة الضغط على غير المفاتيح المطلوبة.
١٠- تستخدم البرمجية إمكانيات الحاسوب بشكل جيد.

أولا : معيار خصائص المحتوى

١- تتبنى البرمجية نظريات تربوية صحيحة في عرضها للمحتوى : ينبغي أن تتبنى البرمجية نظريات تربوية صحيحة ومناسبة للمحتوى عرضا وفلسفة وتقويما. فهل المطلوب أن يقوم الطلاب بعمليات عقلية معقولة ؟ هل هناك مجهودات بذلت للتدرج من الأمثلة المحسوسة لأمثلة أكثر تجريدا ؟ هل نظمت الحقائق بطريقة تساعد الطلاب على الفهم والتذكر، والتحليل والتركيب ... الخ ؟

٢- دقة المحتوى وسلامته العلمية : ينبغي أن يكون محتوى البرمجية دقيقا بأكمله سليما علميا ، وإلا فإن الطلاب سيتعلمون معلومات ومهارات غير صحيحة.

٣- تستخدم البرمجية أنشطة تعليمية مقبولة : ينبغي أن تستخدم البرمجية في طريقة عرضها أساليب وأنشطة تعليمية مناسبة للمحتوى وللطلاب.

٤- تناسب مقدار التعلم مع ما يستغرقه المتعلمون من وقت : ينبغي أن يكون المحتوى غير تافه وذا طبيعة جوهرية ، وألا يستهلك الطلاب أوقات كبيرة مع البرمجية مقابل استفادة قليلة.

٥- وضوح التسلسل والتتابع المنطقي للدروس : ينبغي أن يكون تسلسل الدروس مقبولا ، وأن يكون الشرح مفهوما للطالب المتوسط .

٦- يراعى تحقق الأهداف المذكورة : ينبغي أن تساعد البرمجية الطلاب على إنجاز الأهداف المطلوبة ، وألا يعتمد إنجاز التدريبات على الحظ أو على قدرة الطلاب على استخدام لوحة المفاتيح ؛ أكثر من المهارة الفعلية المطلوب تعلمها ، وأن يصبح الطالب قادرا على تطبيق ما تم تعلمه في مهام تالية.

٧- الاستخدام الملائم للأصوات والألوان : بالطبع يمكن أن تزيد المؤثرات الصوتية والموسيقية من اهتمام الطالب ، ويمكن اعتبارها تغذية راجعة جيدة لإثابة الطلاب عن الإجابات الصحيحة والخاطئة على حد سواء . المهم أن تستخدم بقدر مناسب ؛ حيث إن الاستخدام الزائد للمؤثرات الصوتية يمكن أن يعوق تركيز بعض الطلاب ، فالأصوات المرتفعة التي تظهر عند طالب ما قد تشتت انتباه طالب آخر في حالة تركيز. هذا ويفضل أن تتيح البرمجية للطالب أن يختار بين استعمال الأصوات أو عدم استعمالها. وفي كثير من الأحيان تكون الألوان الجذابة سببا في جعل البرمجية أكثر فاعلية وجاذبية ، وفي أحيان أخرى يتسبب التغيير السريع للألوان أو استخدام ألوان غير واقعية في إعاقة التعلم.

٨- إمكانية طبع أي جزء من المحتوى : ينبغي أن يتوفر في البرمجية إمكانية طباعة جزء أو أجزاء محددة من المحتوى حسب رغبة الطالب أو المعلم للاستفادة منها في المراجعة أو زيادة في التوضيح.

٩- الاستخدام الملائم للرسوم والنماذج المتحركة ولقطات الفيديو: إنه من المفيد تعزيز البرمجية - كما هو الحال في المؤثرات الصوتية المناسبة والألوان الجذابة - بالصور والخرائط والمخططات البيانية والصور المتحركة ولقطات الفيديو ؛ حيث إن للأشكال المتحركة جاذبية خاصة تعمل على زيادة دافعية الطلاب.

١٠- الترابط بين أسلوب التمثيل وحركة الرسوم والنماذج بأهداف المحتوى ومضمونه : ينبغي أن يتوفر في البرمجية نوع من الترابط أو التناغم بين أسلوب التمثيل أو العرض المستخدم وحركة الرسوم والنماذج ، وأن تكون مناسبة لأهداف المحتوى ومضمونه.

ثانيا : معيار خصائص استخدام الطالب

تركز خصائص استخدام الطالب علي نواحي القوة والضعف للبرمجية فيما يتعلق باستخدام الطالب ، هذا وينبغي الاهتمام بالكيفية التي تقاس بها هذه المعايير ، فعلى سبيل المثال ، قد يكون لمعيار تحكم الطالب في اختيار الدرس وزن بسيط بالنسبة لمعلم يرغب في التحكم بنفسه في مواقف التدريبات والممارسة ؛ وفيما يلي نقدم هذه المعايير:

١- لا تتطلب معرفة مسبقة للطالب بالحاسوب : بالتأكيد ستكون البرمجية ذات قيمة أعلى إذا ما استطاع الطلاب تحميلها وتشغيلها بدون تدخل من المعلم ، وبدون الحاجة إلى تدريبات مطولة ، كما ينبغي أن تنتقى المفردات اللغوية المستخدمة في البرمجية ، مع مراعاة التقليل من التعبيرات التي تشتمل على مصطلحات فنية صعبة أو غير مألوفة أو غير مستخدمة من قبل.

٢- حث الطلاب على التعاون والعمل المشترك : تشجيع الطلاب على التعاون فيما بينهم من أجل التوصل إلى حل تمرين ما في الرياضيات أو تحديد موقع ما على خريطة محددة في الجغرافيا ، يعد إضافة جديدة لإثارة الدافعية عند الطلاب وتفاعلهم الاجتماعي.

٣- **لا تتطلب البرمجية من الطالب الرجوع لدليل التشغيل** : يفضل أن تعرض البرمجية جميع التعليمات اللازمة على الشاشة بحيث يكون من السهل فهمها بالنسبة للطلاب. وهذا لا يمنع وجود بعض المواقف التي تتطلب وجود كتيبات عمل ، حيث تتطلب البرمجية الرجوع إلي صفحات معينة من هذه الكتيبات التي تشتمل على رسومات أو خرائط أو معلومات معينة.

٤- **توفر للطالب ملخصا عن أدائه** : ينبغي أن توفر البرمجية إمكانية إعطاء تقارير كاملة عن أداء الطلاب ، على شكل نسب مئوية أو رسوم بيانية، مما يساعد على إثارة الدافعية عند الطالب ، وأن تعطى صورة واضحة للمعلم عن أداء طلابه.

٥- **تغذية راجعة فعالة للاستجابات الصحيحة والخاطئة على حد سواء:** يثير التعزيز الموجب دافعية الطلاب ؛ كما تسهم التغذية الراجعة للاستجابات غير الصحيحة في عملية التعلم بشرط ألا تؤذى مشاعر الطالب، هذا ويفضل استخدام اسم الطالب ، وكذلك التعزيز غير اللفظي : كالمؤثرات الصوتية ، أو الرسوم المتحركة ، أو عرض الموضوعات التي أتقنها الطالب . قد يكون من المفيد أحيانا أن يوجه الطالب إلى معلم الفصل من خلال التغذية الراجعة أو تقديم بعض التوضيحات المساعدة.

٦- **التغذية الراجعة الموجبة أكثر جاذبية من التغذية الراجعة السالبة :** ينبغي أن يكون التعزيز المعطى للاستجابات الصحيحة أكثر جذبا من التغذية الراجعة السالبة التي تعطى للاستجابات غير الصحيحة ، وإلا فإن التلاميذ سيتعمدون الاستجابة الخاطئة من أجل رؤية منظر جذاب لا يعرض عليهم في حالة الاستجابة الصحيحة : فعلى سبيل المثال عندما تعطى البرمجية تعزيزا موجبا حالة الاستجابة الصحيحة مثل : **ممتاز ، الإجابة صحيحة** ، وفى حالة الاستجابة الخاطئة يعرض شكل كاريكاتير عابس فإن الطلاب من المحتمل أن يتعمدوا الاستجابة الخاطئة ، حتى يتمكنوا من رؤية شكل كاريكاتيري آخر.

٧- **تتيح للطالب أن يتحكم في معدل عرض المعلومات** : ينبغي أن تتيح البرمجية للطالب قراءة المعلومات طبقا لسرعته الخاصة في القراءة ، إنه من المحبط جدا اختفاء النصوص الهامة من على شاشة الحاسوب قبل أن ينتهي الطالب من قراءتها وفهمها.

٨- **تتضمن وظائف لتحليل أخطاء الطلاب :** ينبغي أن يتوفر في البرمجية بعض الوظائف التي تتيح للمعلم أن يحصل علي تحليل كامل لأخطاء طلابه ، لتحديد نقاط القوة ونقاط الضعف لأدائهم، وبالتالي يتمكن من إثرائهم أو علاج قصورهم.

٩- **تتيح للطالب أن يتحكم في تسلسل محتويات الدرس :** ينبغي أن تتيح البرمجية للطالب أن يختار ما يفضل أن يتعلمه طبقا لحاجته الفردية : كأن يختار تدريبات وتمارين معينة من الدرس مثلا.

١٠- **تتيح للطالب أن يتحكم في اختيار الدرس :** ينبغي أن تتيح البرمجية للطالب أن يختار الدرس الذي يرغب في تعلمه ، وذلك من خلال توفير قائمة بالدروس التي تحتويها البرمجية ، هذا من شأنه أن يشعر الطالب المستخدم بارتياح عندما يسمح له بتحديد نوع التمارين والتدريبات التي ستقدم له بدلا من ترك التحكم بالكامل للبرمجية نفسها.

١١- **تتيح للطالب أن يختار العودة لمراجعة أجزاء معينة من درس معين :** ينبغي أن تتيح البرمجية للطالب العودة لمراجعة بعض المعلومات السابقة ، فمثلا إذا حدث ووجد الطالب نفسه غير ملم بتعليمات سؤال ما ، فمن المفيد أن يعاود مراجعة الشاشات السابقة لمراجعة ما يحتاجه من تعليمات.

١٢- **تتيح للطالب أن يختار أنماطا مختلفة للعرض :** يفضل الطلاب رؤية صيغ ونماذج مختلفة للعرض نظرا لما بينهم من فروق فردية ؛ فإمكانية اختيار طرق مختلفة للتدريب على المهارات والمفاهيم يمكن أن يكون له بعض الإيجابيات في تعلم الطلاب : فقد يفضل بعض الطلاب من خلال برمجية لتعليم الهجاء مثلا : التدرب على الهجاء بواسطة الكلمات المبعثرة ، في حين قد يفضل غيرهم التعرف على الكلمات من سياق نص ما ، بينما يستمتع آخرون بالتدريب على الكلمات من خلال الألعاب والألغاز.

١٣- **تتضمن البرمجية عدة مستويات من الصعوبة والسهولة :** ينبغي أن تتيح البرمجية للطالب أن يختار مستويات الصعوبة والسهولة عند اختياره للتمارين والتدريبات طبقا لقدراته وإمكاناته.

١٤- سهولة قراءة النصوص المعروضة على الشاشة باستخدام حروف ذات أحجام مناسبة : ينبغي أن تكون العروض والنصوص المعروضة على شاشة الحاسوب مقروءة ، وبحروف مناسبة الحجم .

١٥- تتضمن البرمجية وظائف مساعدة *ON-Line Help* : ينبغي أن يتوفر في البرمجية مساعدة فورية ، يمكن للطالب الحصول عليها بسهولة حالة حاجته إليها.

١٦- التقليل من الاعتماد على المعلم : ينبغي أن تتيح البرمجية للطالب فرصة الاعتماد الكامل علي نفسه ، والذي من شأنه أن يقلل اللجوء للمعلم قدر المستطاع.

ثالثا : معيار خصائص استخدام المعلم

سوف نستعرض فيما يلي أهم المعايير التي تميز البرمجيات الجيدة عن غيرها والتي تسهل عمل المعلم في إدارة عمليتي التعليم والتعلم بمساعدة هذه البرمجية :

١- عرض الأهداف التعليمية بوضوح : ينبغي أن تذكر الأهداف التعليمية في البرمجية نفسها أو في الأدلة المصاحبة لها على الأقل ، حيث نعلم أهمية وجود الأهداف للمعلم والتلميذ على السواء.

٢- تتكامل الأهداف مع المحتوى : ينبغي أن تكون موضوعات البرمجية وما تحتويه من دروس وتدريبات متناسبة ومتمشية مع ما تم سرده من أهداف.

٣- تتيح للمعلم أن يتحكم في مستويات صعوبة بعض الصياغات : ينبغي أن تتيح البرمجية للمعلم التحكم في مستويات صعوبة بعض الصياغات، لتصبح البرمجية أكثر مناسبة لأغلبية فئات الطلاب المتفاوتين في القدرات والمواهب والمهارات.

٤- تتيح للمعلم أن يغير من قوائم المفردات كالكلمات والمسائل : ينبغي أن تتيح البرمجية للمعلم تغيير قوائم المفردات كالكلمات الهجائية (قوائم الحصيلة اللغوية) والمسائل الرياضية ؛ كي لا تقل قيمة البرمجية عندما يتعلم معظم الطلاب الكلمات الهجائية والمسائل الرياضية التي تشتمل عليها البرمجية.

٥- توفر كتيبات للمعلم أو مواد تعليمية مساعدة : ينبغي أن يكون مرفقا مع البرمجية مواد تعليمية مساعدة للمعلم : كدليل المعلم وبعض الكتيبات التي قد تحتوى على بعض الشروح الإضافية أو الإشارة إلى المعلم بأهمية الرجوع إلى مصدر تعليمي آخر ، أو اقتراحات خاصة بما ينبغي أن يتبعه المعلم في إدارة عملية التعلم.

٦- توضح دور المعلم بوضوح : ينبغي أن يوضح في البرمجية تحديدا واضحا لدور المعلم عند استخدام هذه البرمجية داخل حجرات الدراسة.

٧- تقترح خططا للتدريس : ينبغي أن يقترح كتاب المعلم المرفق مع البرمجية خططا للدروس التي تستخدم البرمجية ، وهنا ينبغي التأكد من جودة تلك الخطط بالإضافة إلى حقيقة توفرها في البرمجية.

٨- توفر كراسات عمل مفيدة للطالب : ينبغي توفر كتيبات عمل خاصة بالطلاب ، تشتمل على تدريبات إضافية اختيارية لتوفير وقت الحاسوب ، وقد تحتاج بعض التمرينات والتدريبات الموجود بالبرمجية بعض الخطوات التي ينبغي على الطالب تنفيذها خارج بيئة البرمجية أثناء التعامل معها .

٩- توفر أنشطة إثرائية للطالب سريع التعلم : ينبغي أن تتضمن البرمجية أنشطة إثرائية للطلاب سريعي التعلم ؛ عندما يتقنوا ما تحتويه البرمجية من دروس.

١٠- توفر أنشطة علاجية للطالب بطيء التعلم : ينبغي أن تتضمن البرمجية أنشطة علاجية للطالب بطيء التعلم ؛ الذي سوف يواجه بعض الصعوبات عند تعامله مع دروس البرمجية ، تساعده على إتقانها.

١١- **توفر ملخصا لأداء كل طلاب الفصل** : ينبغي أن توفر البرمجية ملخصات لأداء الطلاب بصورة فردية أو (و) جماعية لتساعد المعلم على الوقوف الكامل على أداء طلابه ، وكذا التخطيط الجيد للدروس التي سيتم عرضها من خلال تلك البرمجية.

١٢- **تقبل البرمجية وتقدم أجوبة متنوعة** : ينبغي أن تكون البرمجية مصممة بحيث تقبل وتقدم نماذج مختلفة للأجوبة الصحيحة .

١٣- **تقترح استخدام أنشطة ومصادر تعليمية أخرى** : ينبغي أن توفر البرمجية أنشطة تعليمية ومواد أخرى إضافية يمكن للطالب والمعلم على حد سواء الاستفادة منها عند الرجوع إليها.

١٤- **إمكانية طبع النتائج المسجلة** : ينبغي أن توفر البرمجية إمكانية طباعة النتائج المسجلة .

١٥- **إمكانية توليد مفردات الاختبارات وطباعتها** : ينبغي أن يتوفر في البرمجية إمكانية توليد مفردات مختلفة ومتكافئة لمفردات الاختبارات (والتي تعرف بالتوليد العشوائي لمفردات الاختبارات) ؛ حيث إنه إذا عرضت مفردات الاختبارات في تسلسل ثابت ، فقد يبدأ الطلاب في الإجابة الصحيحة عن الأسئلة بسبب الترتيب الذي ظهرت به، وليس بسبب حدوث تعلم المهارة أو الحقيقة، أما إذا كانت مفردات الاختبار قد اختيرت بطريقة عشوائية ؛ فإن الترتيب قد لا يعطى أي تلميحات عن الإجابة.

رابعا : معيار خصائص تشغيل البرمجية

سوف نستعرض فيما يلي أهم المعايير التى تميز البرمجيات التعليمية الجيدة عن غيرها فيما يتعلق بتشغيلها :

١- **سهولة الدخول إلى البرمجية والخروج منها** : ينبغي أن توفر البرمجية دخولا سهلا إلي دروسها ، وخروجا منها في أي موقع يكون فيه الطالب ويرغب في الخروج منها.

٢- وجود دليل استخدام البرمجية بصياغة واضحة : تختلف طريقة استخدام البرمجيات من برمجية إلي أخري ، حيث يرجع ذلك إلى مصمم البرمجية وتصوره عن الكيفية التي ستتم بها الاستجابة ، والكيفية التي يتم بها التعامل معها ، وما لم تكن هناك تعليمات واضحة في كتيبات المعلومات المصاحبة للبرمجية ، فإن المعلمين والطلاب قد يجدون أنفسهم مطالبين بالتخمين عند تشغيل البرمجية.

٣- ترابط عرض دروس البرمجية على الشاشة مع المضمون : ينبغي أن يتكامل العرض على الشاشة مع مضمون الدروس التي تحتويها البرمجية.

٤- التنسيق على الشاشة واضح وجميل : ينبغي أن تكون العروض على الشاشة واضحة وذات تنسيق جميل ، يساعد على جذب انتباه التلميذ ، ويعمل علي زيادة دافعيته للتعلم. فينبغي ألا تعرض علي الشاشة نصوص كثيرة ساطعة ، مع ضرورة العمل على انقرائية النص باستخدام الرسومات التوضيحية مع العمل علي توضيح الكلمات الهامة بشكل ما ، وترك فراغات كافية بين الأسطر ، واستخدام حركة النصوص وإضافة بعض الكلمات المومضة ، بشرط عدم المغالاة في ذلك.

٥- تتيح للمستخدم تصحيح أخطاء الكتابة : عندما يقوم المستخدم للبرمجية بكتابة أية معلومات باستخدام لوحة المفاتيح ، فمن المحتمل أن يقع الفرد في خطأ أثناء عملية الكتابة ، وعليه فإنه من الضروري أن تتوفر خاصية التصحيح لتلك الأخطاء بمجرد تداركها وقبل تغذية الحاسوب بها ، ويمكن التأكد من ذلك باستخدام مفاتيح الأسهم : ⇦ ⇧، ⇩ ، ⇨ أو مفتاح المسح *Del* أو مفتاح الرجوع إلى الخلف *Backspace* ⇦ لترى إذا كان من الممكن مسح الأخطاء مما يسمح للمستخدم إعادة إدخال الاستجابة بالطريقة الصحيحة التي يرغب فيها.

٦- سهولة استخدام البرمجية : ينبغي أن تكون البرمجية مصممة بحيث يتقدم الطالب داخل موضوعاتها وأنشطتها من خلال قوائم أساسية وفرعية متسلسلة ، مع مراعاة البساطة في الأسلوب الذي يحدد به الاختيار مثل استخدام مفاتيح الأسهم أو الوظائف أو اختيار رقم أو حرف معين من

القائمة أو عن طريق التأشير بالماوس علي موقع محدد بالشاشة. ومن النقاط التي يجب وضعها في الاعتبار في هذا الصدد : أن تتيح البرمجية للطالب أن يستخدم اختصارات الاستجابات الشائعة مثل (ن) بدلا من (نعم) ، كما ينبغي ألا تتسبب إضافة علامة ترقيم أو فراغ زائد عند الاستجابة في طلب إعادة الاستجابة مرة أخرى ، كما ينبغي ألا يتطلب الانتقال من جزء بالبرمجية إلي آخر ضرورة إيقاف الحاسوب وإعادة تشغيله.

٧- **تتيح اختيار أجزاء محددة من المحتوى :** قد يفضل الطالب في بعض الأحيان أن يبدأ تعلمه بجزء دون غيره ، أو قد يستخدم الطالب الدرس أكثر من مرة فيتقن بعض أجزائه دون غيرها ، والبرمجيات الجيدة هي التي تتيح الفرصة للطالب أن يختار أجزاء محددة من المحتوى ، وتخطى بعضها.

٨- **نصوص البرمجية سليمة اللغة واضحة المعنى :** قد لا تتبع بعض البرمجيات قواعد النحو السليمة ، وقد تشتمل على أخطاء هجائية عديدة أو استخدامات خاطئة لعلامات الترقيم .

٩- **تتيح البرمجية تشغيلا موثوقا** ، وذلك بعدم تعطيلها حالة الضغط على غير المفاتيح المطلوبة : قد تتوقف البرمجية غير الجيدة عن العمل (تتجمد) عندما يضغط الطالب علي مفتاح غير مطلوب بطريق الخطأ ؛ فمثلا إذا كان سؤال ما يتطلب الإجابة عنه بـ (نعم / لا) واستجاب الطالب بالضغط على أي مفتاح آخر فإن البرمجية تتوقف عن العمل ، وهذا يسبب كثيرا من الإحباط للطالب ، وعادة ما يتطلب ذلك إعادة الدرس من بدايته مرة ثانية.

١٠- **تستخدم البرمجية إمكانيات الحاسوب بشكل جيد :** ينبغي أن تستفيد البرمجية الجيدة من إمكانيات الحاسوب المتعددة ، مما يوفر لها مميزات كثيرة، مقارنة بالكتب وبقية المصادر الأخرى ، حيث تضيف تلك الإمكانات أبعادا جديدة لعملية التعليم والتعلم.

قائمة بنود لتقييم وتقويم البرمجيات التعليمية
COURSEWARE

● وصف البرمجية

اسم البرمجية : ....

الناشر : ....   سنة النشر : ....

النمط : ....   المادة المبرمجة : ....

الموضوع : ....   عدد الدروس : ....

المستوى الصفي : ....

● متطلبات التشغيل

نوع الحاسوب المطلوب : ....   سعة الذاكرة المطلوبة : ....

بطاقة الإظهار المطلوبة : ....   متطلبات إضافية : ....

بيئة التشغيل :   ☐ دوس   ☐ ويندوز

إمكانية التعامل مع شبكة الإنترنت :   ☐ نعم   ☐ لا

● سياسة النسخ الاحتياطية

☐ مقبولة   ☐ غير مقبولة

اســـــــم المقيم : ....

صفته (وظيفته) : ....

خصائص المحتوي

م	الخاصية	درجة توفر الخاصية			
		ممتازة (٣)	جيدة (٢)	ضعيفة (١)	لا تنطبق (٠)
١	تتبنى البرمجية نظريات تربوية صحيحة في عرضها للمحتوى.				
٢	دقة المحتوى وسلامته العلمية.				
٣	تستخدم أنشطة تعليمية مقبولة.				
٤	تناسب مقدار التعلم مع ما يستغرقه المتعلمون من وقت				
٥	وضوح التسلسل والتتابع المنطقي للدروس.				
٦	تراعي تحقق الأهداف المذكورة.				
٧	الاستخدام الملائم للأصوات والألوان.				
٨	إمكانية طبع أي جزء من المحتوى .				
٩	الاستخدام الملائم للرسوم والنماذج المتحركة ولقطات الفيديو.				
١٠	الترابط بين أسلوب التمثيل وحركة الرسوم والنماذج بأهداف المحتوى ومضمونه.				

الدرجة الكلية لمعيار خصائص المحتوى (علما بأن الحد الأقصى للنقاط الممكنة ٣٠) = ...
النسبة المئوية لدرجات معيار خصائص المحتوى = ... %

☺ خصائص استخدام الطالب

م	الخاصية	درجة توفر الخاصية			
		ممتازة (٣)	جـيدة (٢)	ضعيفة (١)	لا تنطبق (٠)
١	لا تتطلب معرفة مسبقة بالحاسوب.				
٢	تحث الطلاب على التعاون والعمل المشترك.				
٣	لا تتطلب من الطالب الرجوع لدليل التشغيل.				
٤	توفرللطالب ملخصا عن أدائه.				
٥	تغذية راجعة فعالة للاستجابات الصحيحة والخاطئة على حد سواء.				
٦	التعذية الراجعة الموجبة أكثر جاذبية من التغذية الراجعة السالبة .				
٧	تتيح للطالب أن يتحكم في معدل عرض المعلومات.				
٨	تتضمن وظائف لتحليل أخطاء الطلاب.				
٩	تتيح للطالب أن يتحكم في تسلسل محتويات الدرس.				
١٠	تتيح للطالب أن يتحكم في اختيار الدرس.				
١١	تتيح للطالب أن يختار العودة لمراجعة أجزاء معينة من درس معين.				
١٢	تتيح للطالب أن يختار أنماطا مختلفة للعرض.				

(تابع) خصائص استخدام الطالب

م	الخاصية	درجة توفر الخاصية			
		ممتازة (٣)	جيدة (٢)	ضعيفة (١)	لا تنطبق (٠)
١٣	تتضمن عدة مستويات من الصعوبة والسهولة .				
١٤	سهولة قراءة النصوص المعروضة على الشاشة .				
١٥	تتضمن وظائف مساعدة *On Line Help*				
١٦	التقليل من الاعتماد على المدرس .				

الدرجة الكلية لمعيار استخدام الطالب (علما بأن الحد الأقصى للنقاط الممكنة ٤٨) = ...
النسبة المئوية لدرجات معيار استخدام الطالب = ... %

خصائص استخدام المعلم

م	الخاصية	درجة توفر الخاصية			
		ممتازة (٣)	جيدة (٢)	ضعيفة (١)	لا تنطبق (٠)
١	عرض الأهداف التعليمية بوضوح.				
٢	تتكامل الأهداف مع المحتوى .				
٣	تتيح للمعلم أن يتحكم في مستويات صعوبة بعض الصياغات.				
٤	تتيح للمعلم أن يغير من قوائم المفردات كالكلمات والمسائل.				

(تابع) خصائص استخدام المعلم

م	الخاصية	درجة توفر الخاصية			
		ممتازة (٣)	جـيدة (٢)	ضعيفة (١)	لا تنطبق (٠)
٥	توفر كتيبات للمعلم أو مواد تعليمية مساعدة.				
٦	توضح دور المدرس بوضوح.				
٧	تقترح خططا للتدريس.				
٨	توفر كراسات عمل مفيدة للطالب.				
٩	توفر أنشطة إثرائية للطالب سريع التعلم.				
١٠	توفر أنشطة علاجية للطالب بطيء التعلم.				
١١	توفر ملخصا لأداء كل طلاب الفصل.				
١٢	تقبل البرمجية وتقدم أجوبة متنوعة.				
١٣	تقترح استخدام أنشطة ومصادر تعليمية أخرى.				
١٤	إمكانية طبع نتائج أداء الطلاب.				
١٥	إمكانية توليد مفردات الاختبارات وطباعتها.				

الدرجة الكلية لمعيار استخدام المعلم (علما بأن الحد الأقصى للنقاط الممكنة ٤٥) = ...
النسبة المئوية لدرجات معيار استخدام الطالب = ... %

خصائص تشغيل البرمجية

م	الخاصية	درجة توفر الخاصية			
		ممتازة (٣)	جيـــدة (٢)	ضعيفة (١)	لا تنطبق (٠)
١	ســهولة الدخــول إلــى البرمجيـــة والخروج منها.				
٢	وجـــود دليـــل اســتخدام للبرمجيـــة بصياغة واضحة .				
٣	ترابط عرض دروس البرمجية مـــع المضمون.				
٤	التنسيق على الشاشة واضح وجميل.				
٥	تتيح للمستخدم تصحيح أخطاء الكتابة .				
٦	سهولة استخدام البرمجية .				
٧	تتيح اختيار أجـــزاء مخــددة مـــن محتوى البرمجية.				
٨	نصوص البرمجيـــة ســليمة اللغـــة واضحة المعنى.				
٩	تتيح البرمجية تشغيلا موثوقا .				
١٠	تستخدم إمكانيات الحاســوب بشـــكل جيد.				

الدرجة الكلية لمعيار تشغيل البرمجية (علما بأن الحد الأقصى للنقاط الممكنة ٣٠) = ...
النسبة المئوية لدرجات معيار استخدام الطالب = ... %

القرار النهائي لصلاحية البرمجية

درجة معيار خصائص المحتوى .... = ...
درجة معيار خصائص استخدام الطالب .... = ...
درجة معيار خصائص استخدام المعلم .... = ...
درجة معيار خصائص تشغيل البرمجية .... = ...

المجموع الكلى لنقاط كل الخصائص .... = ...
الحد الأقصى للنقاط الممكنة .... = ١٥٣
التقدير بالنسبة المئوية .... = ... %

التوصية النهائية ☐ تصلح ☐ لا تصلح

الجزء الثاني

إنتاج برمجيات
الوسائط المتعددة التفاعلية

في بيئة فيجوال بيسك

الفصل الخامس

التعرف على بيئة فيجوال بيسك

الفصل السادس

تصميم وإنتاج شاشات النصوص والعرض

الفصل السابع

تصميم وإنتاج شاشات القائمة الرئيسية والقوائم الفرعية

الفصل الثامن

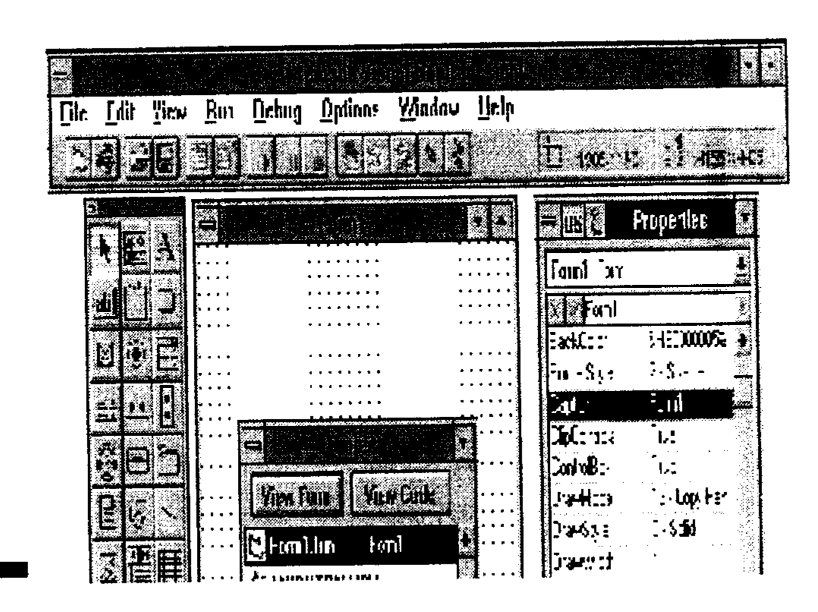
تصميم وإنتاج شاشات التمارين والتدريبات والاختبارات

الفصل التاسع

التعامل مع الصور وتحريكها

الفصل العاشر

كيفية الانتقال من شاشة إلى أخرى

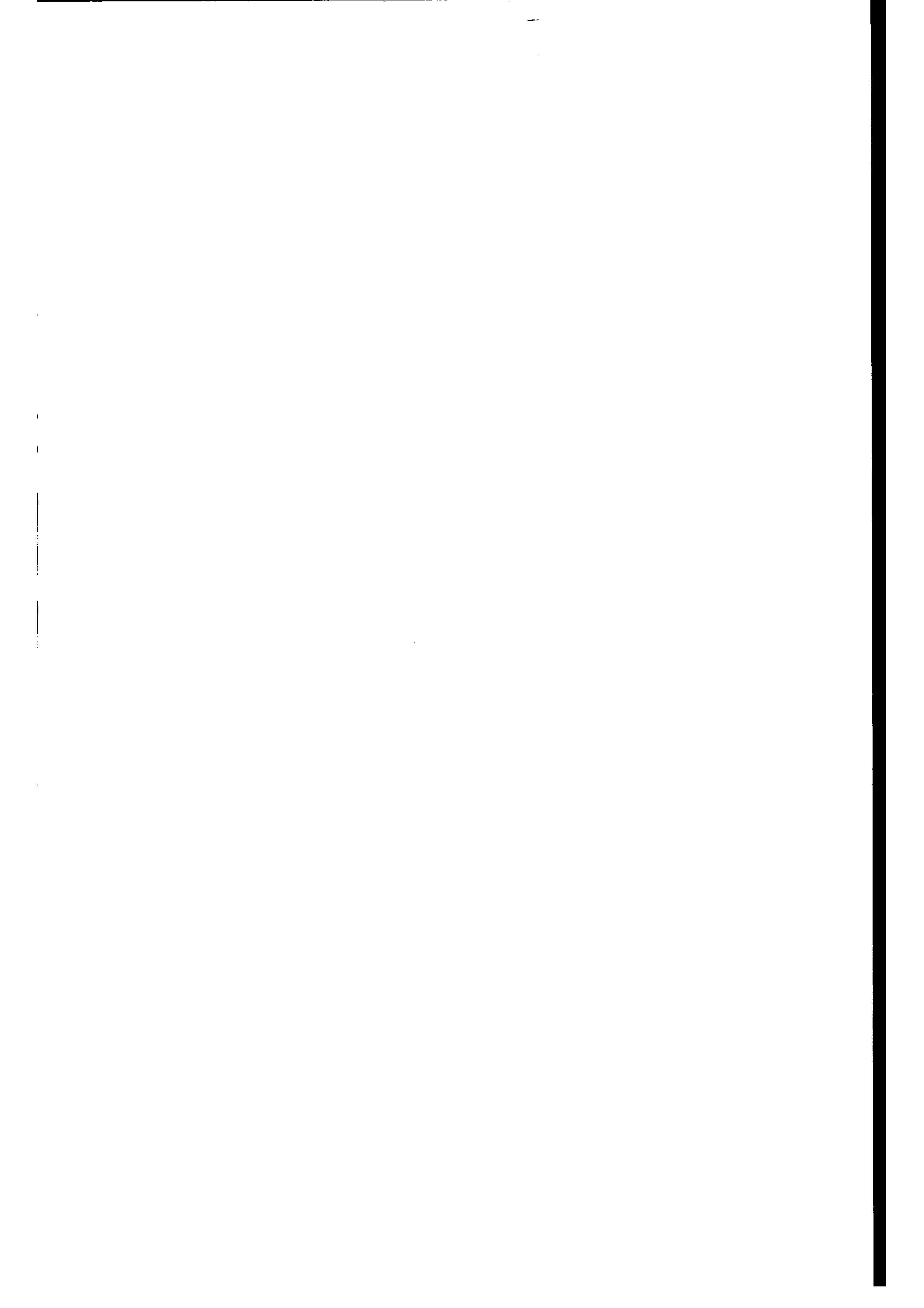


الفصل الخامس

التعرف

على بيئة فيجوال بيسك

Visual BASIC *(VB)*

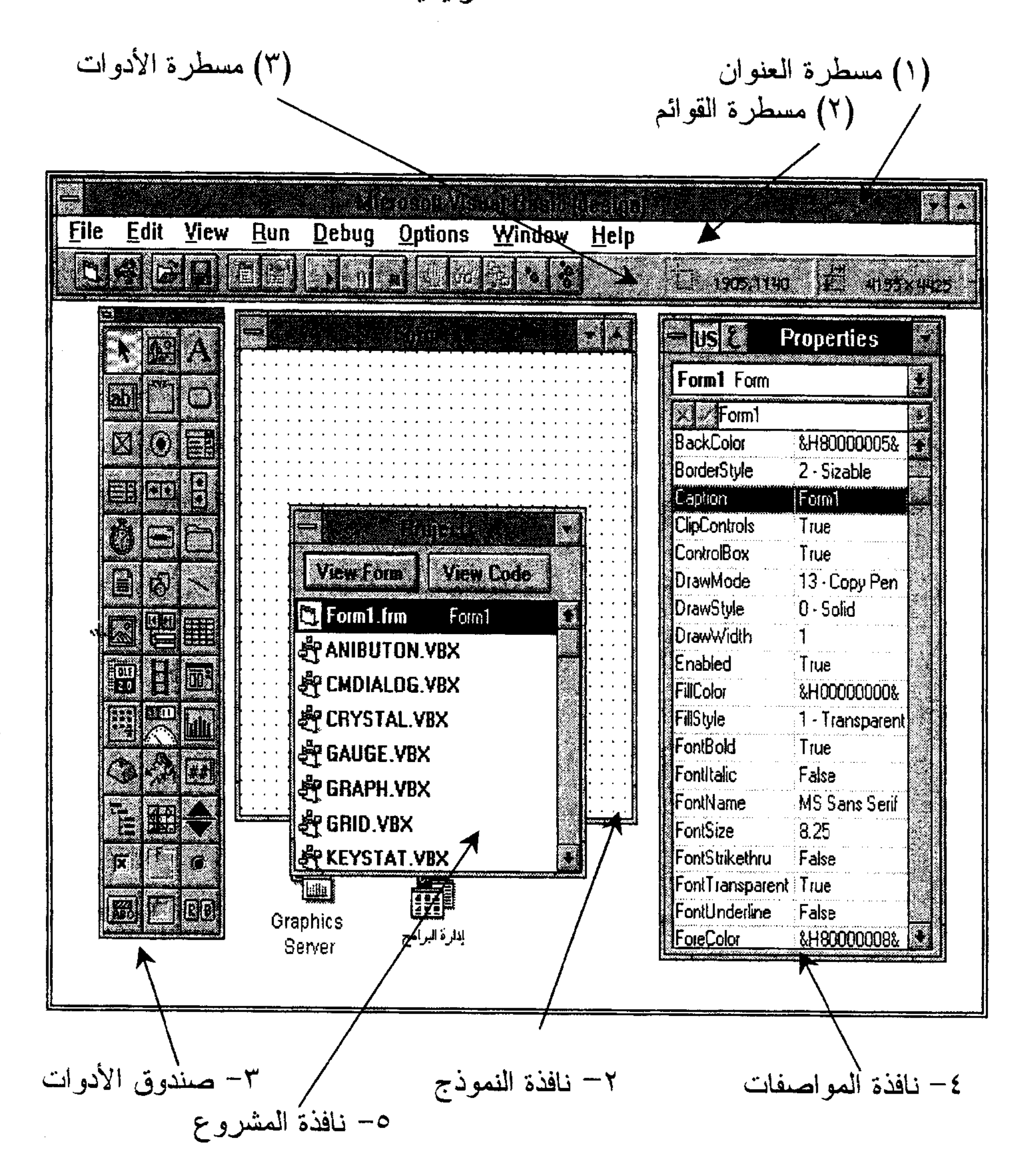


تمهيد

تعتبر لغة البيسك المرئي المدعمة بالرسومات والصــور والتــي تعمـل ضمن بيئة ويندوز *Windows* من ضمن الأنظمة الخاصة (النــوع الثاني) للتأليف التعليمي ، حيث تقدم العديد من التســهيلات التـي تمكن المبرمج من إنتاج الدروس التعليمية كما تتيح له الاستفادة من إمكانية الاستفادة من إمكانات بيئة ويمـب *WIMP* المتوفـرة فـي ويندوز إضافة إلى الخطوط العربية. ورغم مزاياها المتعــددة عـن لغـات البرمجة العامة الأخرى لم تعفي المؤلف تماما من مهمة البرمجة ، وبالتـالي فإن فعلى المصمم أو المبرمج أن يعد الدرس التعليمي فـي صـورة تطبيـق *Application* والذي يتكون عادة من مجموعـة نوافـذ *Forms* وموديـولات *Modules* وبناءات منطقية *Codes* تعمل معا كبرنامج واحد. وعليه فقد يحتاج المبرمج إلى خبره سابقة في البرمجة ليتمكن من بنـاء البنـاءات المنطقيـة (البرنامج) بلغة البيسك المرئي والتي تختلف في بعض المواقف كبيئـة عـن بقية إصدارات البيسك الأخرى (*QBASIC* ، *BASICA* ، *GWBASIC*،...).

تتكون الشاشة الرئيسية لبرنامج البيسك المرئـي (VB) *Visual BASIC* **الإصدار الثالث** من خمس نوافذ رئيسية ؛ قد تظهر متراكبة فـوق بعضـها البعض عند بدء التشغيل إلا أنها قابلة للتحريك من مواقعها وتعديل حجمـها كما هو الحال في تطبيقات النوافذ WINDOWS ، حاول أن ترتبـها لتـأخذ الشكل التالي :

١- القائمة الرئيسية

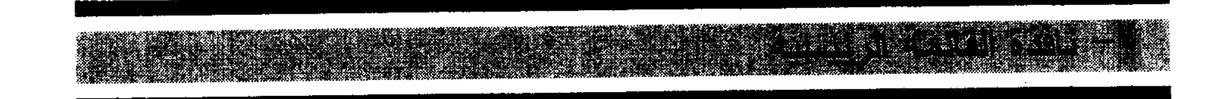


شكل رقم (١)

النوافذ الرئيسية للبيسك المرئي

– نافذة القائمة الرئيسية

وتضم ثلاثة مساطر على النحو التالي :



(١) مسطرة العنوان Title bar

وهى أول مسطرة أعلى النافذة وتحتوى على أسم البرنامج كما تحتوى على ما يلي :

■ قائمة التحكم في البرنامج Program Control Menu

وتوجد على هيئة مربع صغير في أقصى يسار مسطرة العنوان والموضحة بالشكل رقم (٢)

مفتاح التصغير

قائمة التحكم في البرنامج

مفتاح التكبير

شكل رقم (٢)

يوضح مسطرة العنوان

وعند النقر على هذا المربع بالفأرة تظهر لنا قائمة عن طريقها يمكن لنا التحكم في نافذة البرنامج سواء بتحريكها أو إغلاقها نهائيا.

■ مفتاح التكبير Maximize bottom

وهو الموجود في أعلى مسطرة العنوان أقصى اليمين ، وعند النقر عليه بالفأرة فإن نافذة البرنامج أو الملف تأخذ حجم شاشة الحاسوب كاملة ، وعندها يتغير شكله بحيث إذا ما تم الضغط عليه مرة أخرى تعود النافذة إلى حجمها السابق قبل تكبيرها.

■ مفتاح التصغير Minimize bottom

ويوجد في أعلى مسطرة العنوان والى اليسار من مفتاح التكبير السابق شرحه، وعند النقر عليه بالفأرة فإن النافذة المحتوية على البرنامج تتحول إلى رسم تعبيري صغير في أسفل الشاشة ، ويختفي البرنامج تماما من شاشة الحاسوب بينما يظل البرنامج يعمل في ذاكرة الحاسوب وعند اختيار هذا الرسم التعبيري مره أخرى فإن البرنامج يعود إلى شاشة الحاسوب وعلى نفس الهيئة التي تركناه عليها.

(٢) مسطرة القوائم Menu bar

وتوجد أسفل مسطرة العنوان وهى تحتوى ثمانية قوائم منسدلة – سابقة التجهيز – للتحكم في سير البرنامج والوظائف المختلفة التي يتمتع بها البرنامج وهى القوائم الفرعية للتعامل مع الملف File والتعديل Edit والعرض View والتنفيذ Run والتنقيح Debug والخيارات Options النافذة Window والمساعدة Help ، وهى الموضحة بالشكل رقم (٣).

File Edit View Run Debug Options Window Help

شكل رقم (٣)

مسطرة القوائم

قائمة الملف File

وتحتوى على سبعة مجموعات من الإيعازات أو الخدمات:

١- **مجموعة التعامل مع المشروع** : وتتيح البدء في مشروع جديد New Project ، أو فتح مشروع موجود ... Open Project أو حفظه Save Project أو حفظ مشروع تحت مسمى آخر ...Save Project As

٢- **مجموعة التعامل مع النماذج** : حيث تتيح إضافة نموذج معتاد New Form أو نموذج متولد New MDI Form أو نموذج لموديول جديد New Module

٣- **مجموعة التعامل مع الملفات** : حيث تتيح إضافة ملف Add File ... أو حذف ملف Remove File أو حفظ ملف Save File أو حفظ ملف باسم آخر ... Save File As

٤- **مجموعة التعامل مع النصوص** : حيث تتيح تحميل النصوص ... Load Text وحفظها ... Save Text

New Project
Open Project...
Save Project
Save Project As...
New Form
New MDI Form
New Module
Add File... Ctrl+D
Remove File
Save File Ctrl+S
Save File As... Ctrl+A
Load Text...
Save Text...
Print... Ctrl+P
Make EXE File...
1 D:\PROJECT1.MAK
2 CHUNCHER.MAK
3 B:\CRUNCHER.MAK
4 PROJECT1
Exit

٥- **مجموعة الطباعة** ... Print

٦- **مجموعة إنشـــاء ملف تنفيذي** ... Make EXE File يمكن تشغيله من نظام التشغيل الدوس DOS مباشرة.

٧- **الخروج** من البيسك المرئي.

قائمة التعديل Edit

وتحتوى على أربعة مجموعات من الإيعازات أو الخدمات:

Undo Ctrl+Z
Redo
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Link
Delete Del
Find... Ctrl+F
Find Next F3
Find Previous Shift+F3
Replace... Ctrl+R
Bring to Front Ctrl+=
Send to Back Ctrl+-
Align to Grid

١- **مجموعة تعديل ما سبق** وتحتوى علـــى Redo– Undo

٢- مجموعة التعديل الشهيرة والتي تحتـــوى على : **القطع** Cut – **النســـخ** Copy – **اللصق** Paste – **اللصق مـــع الـــتركيب** Paste Link – **المسح** Delate .

٣- **مجموعـــة أدوات البحـــث والاســـتبدال** المألوفة في نظام ويندوز والتي تتضمن علـــى **وظـــائف البحـــث** : ...Find ، Find Next ، Replace ... ، **ووظائف الاستبدال** : .. Replace

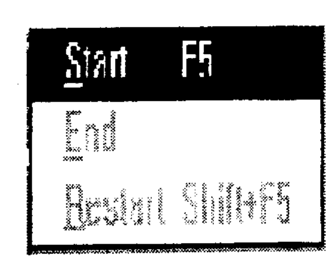
٤- **مجموعة تنظيم وعرض الأغراض** : Bring to Front ، Send to Back ، Align to Grid .

قائمة العرض View

Code F7
New Procedure...
Next Procedure Ctrl+
Previous Procedure Ctrl+
Procedure Definition Shift
√Toolbar

وهى التي تنتج شكل العرض المطلوب حيث يمكن عن طريقها إظهار شاشة محتويات (بناءات) البرنامج أو إظهار مسطرة الأدوات أو التعامل مع الإجراءات أو لإخفائها من الشاشة لإتاحة مساحة أكبر.

قائمة التنفيذ Run



وهى التي تتح للمستخدم تنفيذ المشروع أو البرنامج ، وعادة ما تحتوى قائمة التنفيذ على ثلاثة اختيارات ، والتي تعتمد بالطبع على البرنامج النشط الموجود في بيئة البيسك المرئي ، حيث تتيح هذه القائمة للمستخدم تنفيذ المشروع أو البرنامج Start أو إيقاف التنفيذ End أو إعادة التنفيذ Restart

قائمة التنقيح Debug

Add Watch...
Instant Watch... Shift+F9
Edit Watch... Ctrl+W
Calls... Ctrl+L
Single Step F8
Procedure Step Shift+F8
Toggle Breakpoint F9
Clear All Breakpoints
Set Next Statement
Show Next Statement

وتعرض أربعة مجموعات من الاختيارات أو الإيعازات تساعد المبرمج على تتبع تنفيذ الإجراءات المختلفة من البرنامج واكتشاف مواطن الخلل إن وجدت.

قائمة الخيارات Optionýs Menu

Envlronment...
Project...

والتي عن طريقها يمكن للمبرمج ضبـــط أو تحديد قيم بعض المتغيرات التي تحدد شكل وألوان الأجزاء المختلفة لشاشات التطبيق أو النماذج التـي سيبدأ بها المشروع عند التنفيذ ، وذلك من خـــلال اختيارين أساسيين :

- خيارات البيئة ... Environment
- خيارات المشروع ... Project

قائمة النوافذ Window Menu

Color Palette	
Debug	Ctrl+B
Menu Design	Ctrl+M
Procedures	F2
Project	
Properties	F4
Toolbox	
Data Manager	
Report Designer	

وعن طريقها يمكن أن نغلق أي من النوافذ السبعة المتاحة داخلها وهي :

- الألوان Color Palette
- التنقيح Debug
- تصميم القوائم Menu Design
- الإجراءات Procedures
- المشروع Project
- المواصفات Properties
- الأدوات Toolbox

قائمة المساعدة Help

كالعادة ، وكما في تطبيقات ويندوز يوفر نظام البيسك المرئي مســـاعدة شاملة قائمة على ما يسمى بالنص الفائق *Hyper Text* حيث تحتـــوى قائمـــة المساعدة Help على ثلاثة مجموعات :

١- مجموعة المساعدة الفعلية والتي تحتوى بدورها على مساعدة خاصـــة بالمحتوى *contents* فعند النقر عليها بالفأرة سترى قائمة بكل أقسام المعلومات المتاحة في المساعدة الفوريـــة ، وبـــالنقر بالفأرة على الجمل التي تحتها خط ســينقلك البرنامج إلى الأقسام والموضوعات المعنية التي تحتاج الإطلاع عليـــها ... وأخـــرى للبحث عـــن موضـــــــوع معيـــن ... Search عن طريق كلمة مفتاحيه محـــددة Key word فعند النقر علي الإيعاز ... Search سيظهر لك صندوقـــا حواريـــا لتحدد من خلاله الموضوع المطلوب عن طريق الكلمة المفتاحيـــه ، فينقلـــك البرنامج إلى المعلومات ذات العلاقة بالموضوع المختار ، وثالثـــة لعـــرض إمكانات المساعدة المساندة للبرنامج *Product Support*.

Contents
Search For Help On...
Obtaining Technical Support...
Learning Microsoft Visual Basic
About Microsoft Visual Basic...

٢- مجموعة التعريف بالبرنـــامج Introducing Microsoft Visual Basic وتعليم البرمجة Learning Microsoft Visual Basic

٣- مجموعة عرض بعض المعلومات الأساسية عـــن البرنـــامج About Microsoft Visual Basic

هذا بالإضافة إلى ما يسمى بميزة الإحساس بالنص الفائق ، والذي يعنـــى أنك إذا أردت معرفة المزيد عن أي عنصر ما من عنـــاصر البرنـــامج (زر أوامر ، صندوق حواري ، نافذة ، رسالة خطأ ، ...) فقط اضغط مفتـــاح **F1** وسوف تعرض لك على الفور المعلومات ذات العلاقة بالموضوع.

(٣) مسطرة الأدوات Tool bar

وهى المسطرة الواقعة أسفل مسطرة القوائم ، والتي تتضمن على أربعـــة عشر زر أو أيقونة - شكل تعبيري - بالنقر على أي منها بالفأرة يتم تنفيـــذ الأمر أو الإيعاز المعنى بها. فمثلا لحفظ مشروع ما ... يتم النقر بالفأرة على الشكل التعبيري بدلا من فتح النافذة المنســـدلة الخاصة بالملف File ثـــم اختيار Save ... وهكذا ، والشكل رقم (٤) يوضح مسطرة الأدوات .

شكل رقم (٤)

مسطرة الأدوات

هذا ويوجد في الجهة اليمنى من مسطرة الأدوات منطقتين لإظهار بيانات موقع النموذج أو وحدة وبيانات حجمها.

٢- نافذة النموذج Form

Form1

النموذج هو النافذة التي سيراها المستخدم عند تنفيذ البرنامج ، وعند فتح مشروع برنامج جديد فإن البيسك المرئي ينشئ نموذجا فارغا باسم Form1 ، حيث يكون بمثابة اللوحة التي تضع عليها الأجزاء المختلفة للتطبيق ، هذا ويتكون التطبيق Application من ما يسمى بالأغراض أو وحدات الإدارة Objects كصناديق الصور أو أزرار الاختيارات أو الأعمدة المنزلقة ؛ كما هو موضع بالشكل رقم (٥).

شكل رقم (٥)

نافذة النموذج

صندوق الأدوات

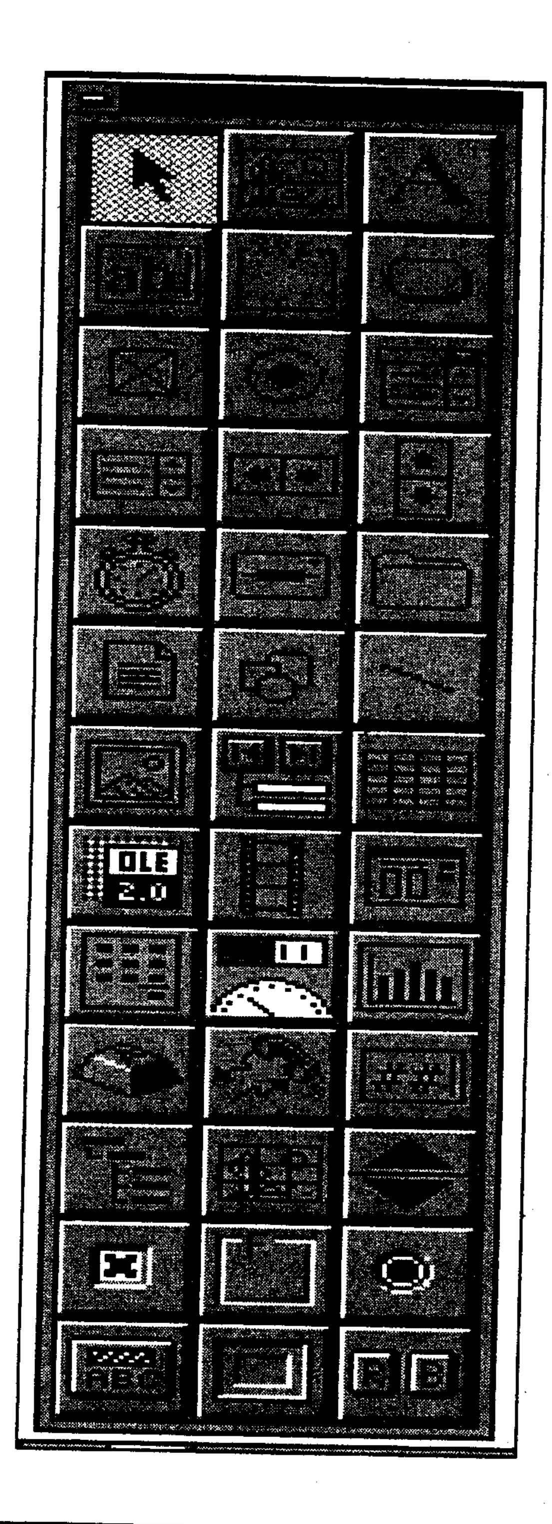
يتم إنشاء **وحدات الادارة Controls** على **نموذج المشروع Form** من خلال **صندوق الأدوات Toolbox** ، حيث إن الأدوات أو الأيقونات الموجودة بصندوق الأدوات ما هي إلا أشكال تعبيرية بواسطتها يمكن تثبيت وحدات الإدارة بالشكل الذي يريده المصمم حسب طبيعة كل أداه أو أيقونة. ويتم ذلك بالنقر على الأيقونة وسحبها إلى نافذة النموذج لتحديد مكانها وحجمها ، والشكل رقم (٦) يوضح صندوق الأدوات ككل ، بينما يوضح الجدول رقم (١) خصائص ووظائف تلك الأدوات.



إن البيسك المرئي – في مرحلة التصميم – يقدم لك الكثير من الأدوات لمساعدتك في إنشاء النموذج الذي سيعرض للمستخدم وفي كتابة سطور البرنامج ، وعلى هذا الأساس يمكنك تغيير حجم الأغراض (النماذج ووحدات الإدارة) ومواقعها أو أي من مواصفاتها الأخرى.



إن البيسك المرئي – في مرحلة التنفيذ – يزيل جميع أدوات التصميم (صندوق الأدوات ، ونافذة المشروع) من على الشاشة ، ويظهر شكل نافذة المستخدم على الشاشة ، حيث يتمكن المستخدم من التفاعل مع البرنامج ويعمل البيسك المرئي على تنفيذ أوامر سطور البرنامج.



شكل رقم (٦)

يوضح صندوق الأدوات ككل

وظائف وحدات التحكم وخصائصها			
الأداة	الافتراضي Default	البادئة Prefix	الوصف Description
المؤشر *Pointer*			يساعد المصمم على تحديد مكان وحجم وحدة الإدارة. هذا ويعتبر المؤشر هو الأداة سابقة التعريف Default ، أى عندما لا يكون الاختيار محددا لأى أداة ، وهو الوسيلة التي يشار بها إلي الأهداف أو الأدوات ، وهو لاينتمى إلى الأدوات ولكنه يستخدم في اختيارها.
صندوق صورة *Picture box*	Picture1	Pic	يساعد المصمم في أن يعرض صورة ، كما يمكن استخدامه كصندوق لوحدات تحكم أخرى.
اللاصق *Label*	Label 1	Lbl	يساعد المصمم في أن يعرض نصا ثابتا كلاصق ؛ أى لا يسمح للمستخدم بتعديل هذا النص مباشرة أثناء تنفيذ البرنامج.

الأداة	الاسم الافتراضي Default	البادئة Prefix	الوصف Description
صندوق نصوص Text box	Text1	Txt	يساعد المصمم في أن يعرض صندوقا لنص يمكن للمستخدم أن يعدله ، كما يمكن أن يكون صندوقا خاليا يسمح للمستخدم بإدخال نص ، هذا ويجب أن يكون حجم الصندوق مناسبا لحجم النص المدخل أو المتوقع إدخاله بواسطة المستخدم.
الإطار Frame	Frame1	fra	يساعد المصمم في أن يرسم إطارا لتنسيق مجموعة من وحدات التحكم بداخله أو التعامل معها كوحدة واحدة عند إعداد الكود ، هذا ويجب رسم الإطار أولا ثم وضع وحدات التحكم بداخله ثانيا.
زر أوامر Command button	Command1	cmd	يساعد المصمم في أن يرسم زرا يمكن للمستخدم أن ينقر عليه لتنفيذ أمر ما. حيث يستجيب هذا الزر للضغط علية بغرض تنفيذ أمر معين أو حدث.

[illegible]	[illegible]	[illegible]	[illegible]
صندوق تعليم (وضع علامة) *Check box*	Check1	chk	يساعد المصمم في أن يرسم زرا يُمكن المستخدم أن يختارما بين حالتين : **التشغيل** أو **التبطيل** لإحدى الخصائص ، وعندما يكون الزر في وضع التشغيل تظهر بداخلة علامة × ، وتتحدد قيمة تلك الخاصية بالقيمة True ، بينما تتحدد قيمة تلك الخاصية بالقيمة False حالة التبطيل.
زر اختيار *Option button*	Option1	opt	يساعد المصمم في أن يرسم زرا للاختيار مثل وحدة التحكم السابقة ، هذا ويجب أن يستخدم ضمن مجموعة أزرار لعرض مجموعة أزرار لعرض مجموعة اختيارات بحيث يسمح للمستخدم أن يحدد خيار واحد فقط. وتظهر بداخلة دائرة صغيرة مصمته عندما يكون عاملا (value=True) وتختفى الدائرة فى حالة التبطيل (value=False).

الأداة	الاسم الافتراضي Default	البادئة Prefix	الوصف Description
صندوق قائمة *List box*	List1	lst	يساعد المصمم في أن يرسم صندوقا لعرض قائمة من العناصر لاختيار أحدها ، ويجوز أن يحتوي الصندوق علي عدة أعمدة ، وإذا زادت محتويات الصندوق عن مساحته فإن قضبان إنزلاق *Vertical scroll bar* رأسية تظهر على الجانب الأيمن للصندوق.
صندوق كمبو *Combo box*	Combo1	cbo	يحمل هذا الصندوق كلا من خاصيتي صندوق النصوص Text Box وصندوق القائمة List Box ، والاسم كمبو مشتق من العبارة Combination Box بمعنى الصندوق المركب ، هذا ويسمح صندوق كمبو للمستخدم أن يحقق اختيارا ما إما بالكتابة أو بإلتقاط الاختيار المطلوب من القائمة (كما في حالة فتح ملف أو حفظ ملف : حيث يمكنك من أن تكتب اسم الملف أو تلتقطه من القائمة) ، وعنما تزيد محتويات الصندوق عن مساحته تظهر على جانبية قضبان انزلاق رأسية.

قضيب إنزلاق أفقي *Horizontal Scroll Bar*	HScroll1	Hsb	يساعد المصمم في أن يرسم قضيب إنـــزلاق أفقـــي يحقـــق للمستخدم إمكانية اختيار مشـــهد معين من النـــافذة فـــى الأتجـــاه الأفقي كما يستخدم لبيان سرعة أو كمية شيء ما.
قضيب إنزلاق رأسي *Vertical Scroll Bar*	Vscroll1	Vsb	يساعد المصمم في أن يرسم قضيب إنزلاق رأســـي يحقـــق للمستخدم امكانية اختيار مشـــهد معين من النـــافذه فـــي الاتجـــاه الرأسي ، كما يســـتخدم لبيـــان سرعة أو كمية شيْء ما.
عداد *Timer*	Timer1	Tmr	يساعد المصمم في أن يتحكم في زمن عرض النوافذ أو لتصيد الأحداث *Event Trapping* فـــي فترة زمنية محددة ويعرض أثناء مرحلة التصميم *Design Time* ولا يعرض بالطبع أثناء مرحلـــة التشـــغيل *Run Time* ، أى أنـــه يمكن ضبط العداد بأن يبدأ حادثة معينة بعد مرور فـــترة زمنيـــة معينة.

الأداة	الاسم الافتراضي Default	البادئة Prefix	الوصف Description
صندوق عرض مشغلات الأقراص *Drive list box*	d: active drive القرص النشط	Drv	يساعد المصمم في أن يرسم صندوقا لعرض قائمة بأسماء بمشغلات الأقراص المتاحة ليختار المستخدم واحدا منها.
صندوق عرض الفهارس *Directory list box*	الفهرس الفرعي النشط Dir1	dir	يساعد المصمم في أن يرسم صندوقا لعرض قائمة للفهارس الفرعية الموجودة بالقرص الحالي لكى يختار المستخدم واحدا منها.
صندوق عرض الملفات *File list box*	لإظهار الملفات الموجودة بالفهرس الفرعي File1	fil	يساعد المصمم في أن يرسم صندوقا لعرض قائمة بالملفات الموجودة علي القرص النشط وبالفهرس العامل لإختيار أحدهما.

[illegible]	Default	Prefix	Description
أداة رسم الأشكال الهندسية *Shape*	Shape1	shp	يمد المستخدم بوسيلة لرسم أشكالا هندسية كالدائرة والمستطيل والمربع والشكل البيضاوي (أشكال هندسية مستوية).
أداة رسم الخطوط *Line*	Line1	lin	تمد المستخدم بوسيلة لرسم خطوطا مستقيمة مختلفة الأحجام ومتنوعة الأشكال.
صندوق صورة مبسطة *Image*	Image1	img	يساعد المصمم في أن يعرض صور مبسطة وفقاً للنماذج *Bitmap* ، *Icon* *Metafile* وهو يشبه في عمله صندوق الصور Picture box

الأداة	الاسم الإفتراضي Default	البادئة Prefix	الوظيفة Description
أداة الإتصال بالبيانات *Data Control*	Data1	dta	وهي أداة تُمكن المستخدم من الإتصال ببعض قواعد البيانات *Ms Access* .
عرض جدولا للبيانات. *Grid*	Grid1	grd	يساعد المصمم في أن يرسم جدولا لعرض البيانات في صورة صفوف وأعمدة.
أداة الربط *OLE*	OLE1	ole	يساعد المصمم في أن يربط صورة أو جدول أو نص ***Object Linking & Embedding*** من تطبيقات وندوز الأخرى الى المشروع الذى يصممه في بيئة البيسك المرئى VB. كما يتيح للمصمم إضافة المؤثرات الصوتية بإستخدام المسجل أو استخدام المكتبة الصوتية لوندوز.

[illegible]	[illegible]	[illegible]	[illegible]
صندوق توليد الحركة *Animoted Botton*	**AniButton1**		يساعد المصمم في أن يضيف الحركة الى تطبيقه.
صندوق حوار *Common dialog*	**CMDialog1**	cdg	يساعد المصمم في أن يرسم صندوقا للحوار يتعامل معه المستخدم لفتح أو تخزين ملفات لإختيار نوع الخط أو اللون.
صندوق تقرير *Crystal custom control*	Report1		يساعد المصمم في أن ينشأ صندوقا يتيح للمستخدم الحصول علي تقرير كامل عن عملة.

الأداة	الاسم الإفتراضي Default	البادئة Prefix	الوصف Description
صندوق الرسوم البيانية *Graph*	Graph1	gph	يساعد المصمم في أن ينشأ صندوقا للرسومات البيانية على إختلاف أنواعها مثل : *Area Line ، 2D and 3D Bar Log/Lin ، 2D and 3D Pie Scatter* ... الخ
صندوق الإتصالات *Communication*	Comm1		يساعد المصمم في أن ينشأ صندوقا يتيح للمستخدم الإتصال بالشبكات المتاحة.
صندوق الصور متعددة المصادر *Picture clip*	PicClip1		يساعد المصمم في أن يجلب بعض الصور من تطبيقات وندوز المختلفة الأخرى الي التطبيق الذى يتعامل معه في بيئة البيسك المرئي.

[illegible]	[illegible]	[illegible]	[illegible]
زر التزييد *Spin button*	Spin1		يساعد المصمم في أن ينشأ مفتاحا (أو زرا) للتزييد : للتزييد أو التنقيص
صندوق تعليم ثلاثي الأبعاد *3D Check box*	**Check3D1**	chk	مثله مثل صندوق التعليم العادي يساعد المصمم في أن يرسم زرا ثلاثي الأبعاد يمكن للمستخدم أن يختاره ليوضح ما إذا كان نص ما صح أو خطأ، هذا ويمكن أن يستخدم أيضا ضمن مجموعة لعرض عدة إختيارات عندما يسمح للمستخدم أن يحدد أكثر من خيار.
إطار ثلاثي الأبعاد *3D Frame*	**Frame3D1**	fra	يساعد المصمم في أن يرسم إطارا ثلاثي الأبعاد لتنسيق مجموعة من وحدات التحكم بداخله أو التعامل معها كوحدة واحدة عند إعداد الكود ، هذا ويجب رسم الإطار أولا ثم وضع وحدات التحكم بداخله ثانيا.

الأداة	الاسم الافتراضي Default	البادئة Prefix	الوصف Description
زر إختيار ثلاثي الأبعاد *3D Option botton*	Option3D1	opt	يساعد المصمم في أن يرسم زرا ثلاثي الأبعاد للإختيار مثل وحدة التحكم السابقة ، هذا ويجب أن يستخدم ضمن مجموعة أزرار لعرض مجموعة إختيارات بحيث يسمح للمستخدم أن يحدد خيارا واحدا فقط.
زر أوامر ثلاثي الأبعاد *3D Command button*	Command3D1	cmd	يساعد المصمم في أن يرسم زرا ثلاثي الأبعاد يمكن للمستخدم أن ينقر عليه لتنفيذ أمر مابحيث يستجيب هذا الزر للضغط عليه بغرض تنفيذ أمر معين أو حدث.
أداة إنشاء بنل *3D Panel*	Panal3d1		يساعد المصمم في أن يعرض نصا ثابتا طلاصق لايسمح للمستخدم بتعديله مباشرة أثناء تنفيذ البرنامج ، وهو يشبه في وظيفته أداة اللاصق ولكنه ذو مظهر أكثر جمالا وجاذبية.

٤- نافذة المواصفات أو الخصائص Properties

تعتبر المواصفات أو الخصائص أسلوب أداء تقنى لوصف صفات وخصائص غرض من الأغراض سواء كان نموذجا أو وحدة إدارة ، وعادة ماتحدد هذه المواصفات بقيم رقمية أو كلمات وصفية من خلال نافذة المواصفات Properties والموضحة بالشكل رقم (٧)

يتم التحكم في مظهر أو أداء كل غرض من الأغراض المستخدمة في مشروع البيسك المرئى بتحديد عدد معين من المواصفات *Sitting* : بعضها يقتصر تحديدها على قيم معينة فمثلا خاصية الرؤية *Visible* لأى غرض يمكن وضعها فقط بحالتى *True* أو *False* .

Form1 Form

A Number Cruncher

AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	A Number Crunc
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparent
FontBold	True
FontItalic	False
FontName	MS Sans Serif
FontSize	8.25
FontStrikethru	False
FontTransparent	True

شكل رقم (٧)

نافذة الخصائص

وهذا يعنى أن الغرض بالنسبة لخاصية الرؤية إما أن يكون ظاهرا أو غير ظاهر، وبعضها الآخر يمكن تحديدة حسب رغبة المصمم او المبرمج فمثلا يمكن تحديد خاصية المسمى *Caption* لنافذة النموذج بأى إسم ، ومما **هو جدير بالذكر أنه ليس مطلوبا منك تحديد قيمة لكل**

خاصـــــــية ، حيث أن أغلب القيم الإفتراضية للعديد من الخواص تكونـن مقبولة أتوماتيكيا *Default* كما هى .

وتتكون نافذة الخصائص كما هو موضح بالشكل رقم (٧) من مسـطرة العنوان وصندوق الأغراض *Objects Box* وصندوق التحديدات *Settings Box* وقائمة الخصائص *Properties List*

مسطرة العنوان Titel bar

وتوجد في أعلى نافذة الخصائص وتشـمل على عنوان النافذة Properties وزر التصغـير Minimize وزر التحكم في النافذه

صندوق الأغراض Objects Box

وهو عبارة عن قائمة منسدلة تعرض أسماء الأغراض التـي يحتويـها التطبيق ، وبصفة مبدئية تأخذ هذه القائمة الشكل التالى :

Form1 Form

وذلك عند بدء الإسـتخدام حيـث تحتوى على معلومات عن النموذج فقط ، ولكن كلما أضفنا وحدة إدارة الـى النموذج فإن البيسك المرئى يقوم بإضافة معلومات عن هذا العرض في تلـك الصندوق.

صندوق التحديدات Settings Box

ومن خلاله يمكن تحديد قيم الخاصية التى تـم إختيارهـا مـن قائمـة الخصائص ، يمكنك تغيير قيمة الخاصية بكتابتـها بنفسـك فـي صنـدوق التحديدات

أو إختيار إحدى القيم المحددة مسبقا من صندوق التحديدات. وذلـــك حســب طبيعة الخاصية .

A Number Cruncher

قائمة الخصائص Properties List

يمكنــك التحــرك خلالهــا لإختيــار الخاصية المطلوبة ، ومن ثـــم تحديــد قيمها بصندوق التحديدات ، وتتكـــون هذه القائمة من مجموعـــة خصـــائص يمكن للمبرمج تعديلها أثنـــاء مرحلـــة التصميم، وهى تختلف فإختلاف نوافــذ النموذج، ووحدات التحكم ، هذا وتوجد الخصائص بالقائمة مرتبة وفقا للترتيب الأبجدى كما هو موضح بالشكل رقـــم (٨).

Properties

Form1 Form

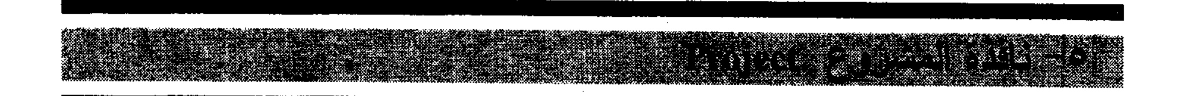
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparent
FontBold	True
FontItalic	False
FontName	MS Sans Serif
FontSize	8.25
FontStrikethru	False
FontTransparent	True
FontUnderline	False

شكل رقم (٨)

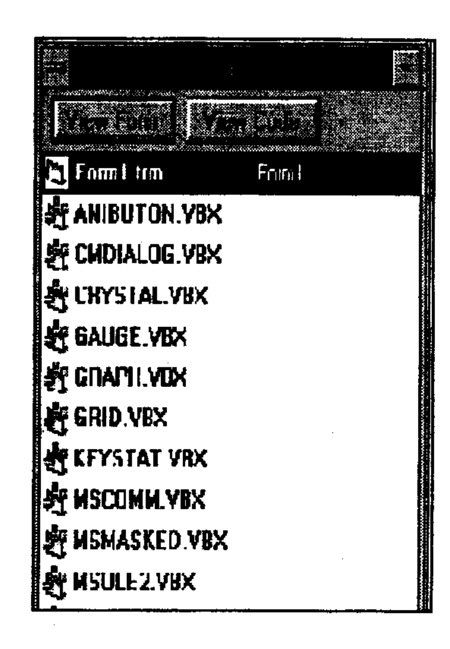
قائمة الخصائص

وينتقى المبرمج الخاصية ثم يحدد القيمة بفتح صندوق التحديدات ، فمثـــلا يتم التحكم في لون الخلفية عن طريق الخاصية BackColor ، ويتم التحكم في نوع الخط عن طريق الخاصية FontName ، ويتم التحكم فـــي إســـــتيراد

صورة عن طريق الخاصية Picture كما يتـــم تحديـــد العنـــوان أو النـــص المعروض عن طريق الخاصية Caption ، ... وهكذا.



وتتكون نافذة المشروع عادة من مسطرتين وقائمة كالتالى :



١- **مسطرة العنوان** : والتى تحمل رقم المشروع وحالته PROJECT1.MAK

٢- **مسطرة العرض** : وتحتوى على :

زر لعرض النموذج **View Form** حيـــث يتم عرض النموذج المقابل لأى مشـــروع (أو ملف) يتم إختياره من نافذة المشروع ، حيـــث يتيح للمبرمج تصميم شاشة الإستخدام اللازمـــة للتطبيق الذى يقوم المصمم بإنشـــائه ، وهـــى عبارة عن الجـــزء (النـــافذه) التـــى يراهـــا المستخدم ويتفاعل معها.

وزر آخر لعرض سطور البرنامج **View Code** حيث يتم عرض ســـطور البرنامج (البناءات المنطقية) المرتبط بالملف الذى تم إختيـــاره مـــن نـــافذة المشروع.

عندما تقوم بإنشاء مشروع من خلال بيئة البيسك المرئي سوف تمر بمرحلتين أساسيتين هما :

١- مرحلة تصميم نماذج المشروع الذى سيراه المستخدم.
٢- مرحلة كتابة سطور البرنامج Coding ، والذى سوف يتم تنفيذ وإدارة المشروع من خلاله.

للعودة مرة أخرى من نافذة سطور البرنامج الى نافذة النموذج يمكنك ببساطة إما النقر بالفأرة على أى جزء ظاهر من النموذج نفسه لتنشيطه أو النقر بالفأرة على زر عرض النموذج **View Form** في نافذة المشروع.

٣- القائمة : وتحتوى على كل الملفات المطلوبه لتنفيذ برنامج البيسك المرئي قيد الإنشاء ، وهما عادة ثلاثة ملفات :

الملف الأول Form1.Frm : ويظهر أمامه عنوانه Form1 الذى يشير الى أن هذا الملف خاص بالنموذج المسمى Form1 ، وهو إسم إعتبارى يمكنك تغييره.

الملف الثانى : GRID.VBX ، **والملف الثالث : OLECLIEN.VBX**

بعد الإنتهاء من عملية تصميم النوافذ Forms بكل محتوياتها يستطيع المبرمج إعداد البناءات المنطقية Cods (أى سطور البرنامج) من خلال نافذة المشروع والتى تعرض محتوياته ، يختار المبرمج النافذه الأولى في المشروع Form1 ثم بالنقر على زر View Code يظهر صندوق حوار يكتب فيه البناءات المنطقية التى تتحكم في النافذة بكل مكوناتها من وحدات تحكم.

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

الفصل السادس

تصميم وإنتاج شاشات النصوص والعرض

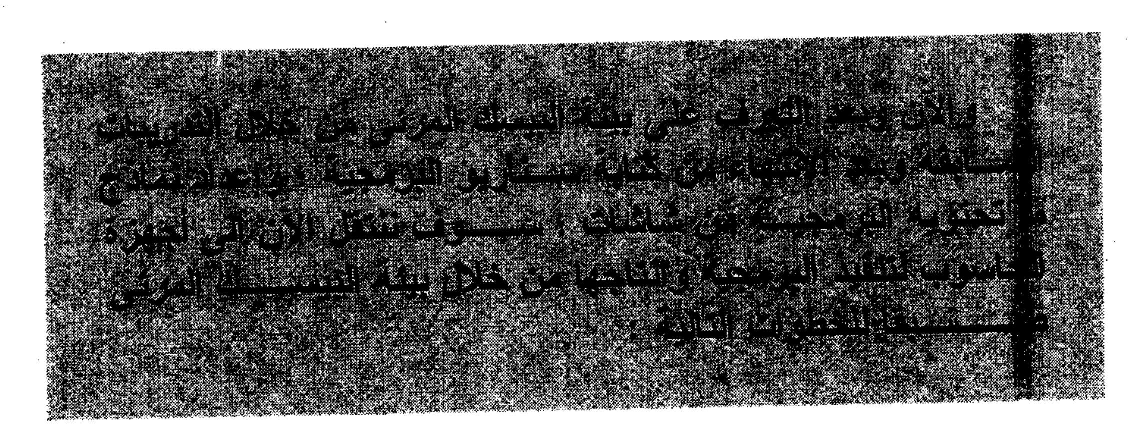
تمهيد

تتكون البرمجية التعليمية عادة من عدة موضوعات ، حيــــث يتكــون الموضوع بدوره من عدة دروس ، ويتكون كل درس من عـــــدة فقرات ، ... وتتكون الفقرة من عدة نوافذ (شاشات) تعرض مــن خلالها المواد التعليمية : فى صورة عروض تدريسية (تدريـــس خصوصي Tutorial) والذى عادة مايتضمن العـــرض Prsentation مدعمـــا بالصور والمؤثرات الصوتية والحركة، والحــوار Dailog وعــرض أمثلـــة وتمارين Examples & Exersices ، وتقديم مفـــردات إختبـــار Test Items: تشخيصية Diagnostic، أو بنائية أو نهائية (إتقان Mastery) ، إضافة إلــى مجموعة ملفات لحفظ أداء المتعلمين.

وفى هذه المرحلة يصمم المبرمج نوافذ التفاعل مع المتعلـــم ويكتــب البناءات المنطقية التى تتحكم فى تتدفــق تلــك النوافــذ ؛ طبقــا ليمــا ورد بالسيناريو.

وعلي المنفذ في مرحلة التشغيل أن يشغل المبرمج المشـــروع بــهدف تصيح الأخطاء البرمجية Debugging .

وعلي المنفذ فى مرحلة التجريب أن يجرب تشغيل التطبيق خارج بيئـــة التصميم والتجريب (بيئة البيسك المرئي) بنفسه أولا ثم مـــع مجموعـــة مــن المتعلمين (أو المتلقين) ثانيا.



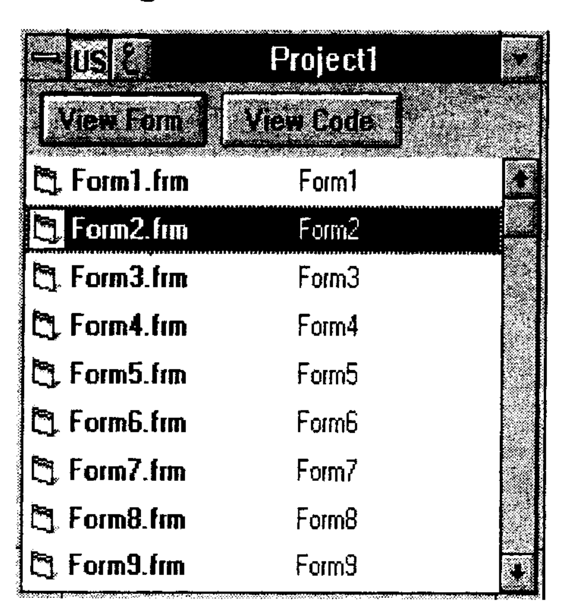
والآن وبعد التعرف على بيئة البيسك المرئي من خلال التدريبات السابقة وبعد الانتهاء من كتابة سيناريو البرمجية ، وإعداد النماذج المحتوية للبرمجية من شاشات ، ســوف ننتقل الآن إلى أجهزة الحاسوب لتنفيذ البرمجية وإنتاجها من خلال بيئة البيســك المرئي طبقــا للخطوات التالية.

أولا : إبدأ مع مشروع جديد New Project

وذلك بفتح النافذة المنسدلة للملفات File وإختر مشروع جديد ، سوف يظهر على الفور شاشة النموذج الأولى Form1

ثانيا : أضف جميع النماذج اللازمة للبرمجية المحددة بالسيناريو

اضف بقية الشاشات اللازمة للبرمجية المحددة بالسيناريو ، بفتح النافذة المنسدلة للملفات File وإختر New Form سوف يظهر على الفور نموذج الشاشة الثانية Form2 كرر هذا العمل حتى تنتهى من إضافة كافة الشاشات اللازمة لمشروعك. سوف تظهر جميع النماذج (الشاشات) التى يتكون منها مشروعك داخل نافذة المشروع .

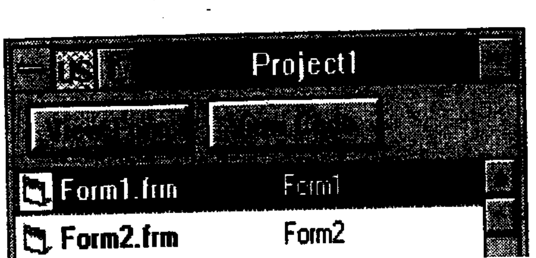


يمكنك اضافة الشاشات السابقة ايضا بالضغط المتكرر على اداة إضافة النماذج بشريط الأدوات

ملاحظة

يمكنك اضافة أى شاشة اضافية لمشروعك فى اى لحظة بأى من الطريقتين السابقتن ، كما يمكنك حذف أى شاشة غير مرغوب فيها من شاشات المشروع ؛ وذلك باختيارك لتلك الشاشة من نافذة المشروع ثم إختيار ***Remove File*** من النافذة المنسدلة File

تصميم شاشات المقدمة وشاشة المنتج

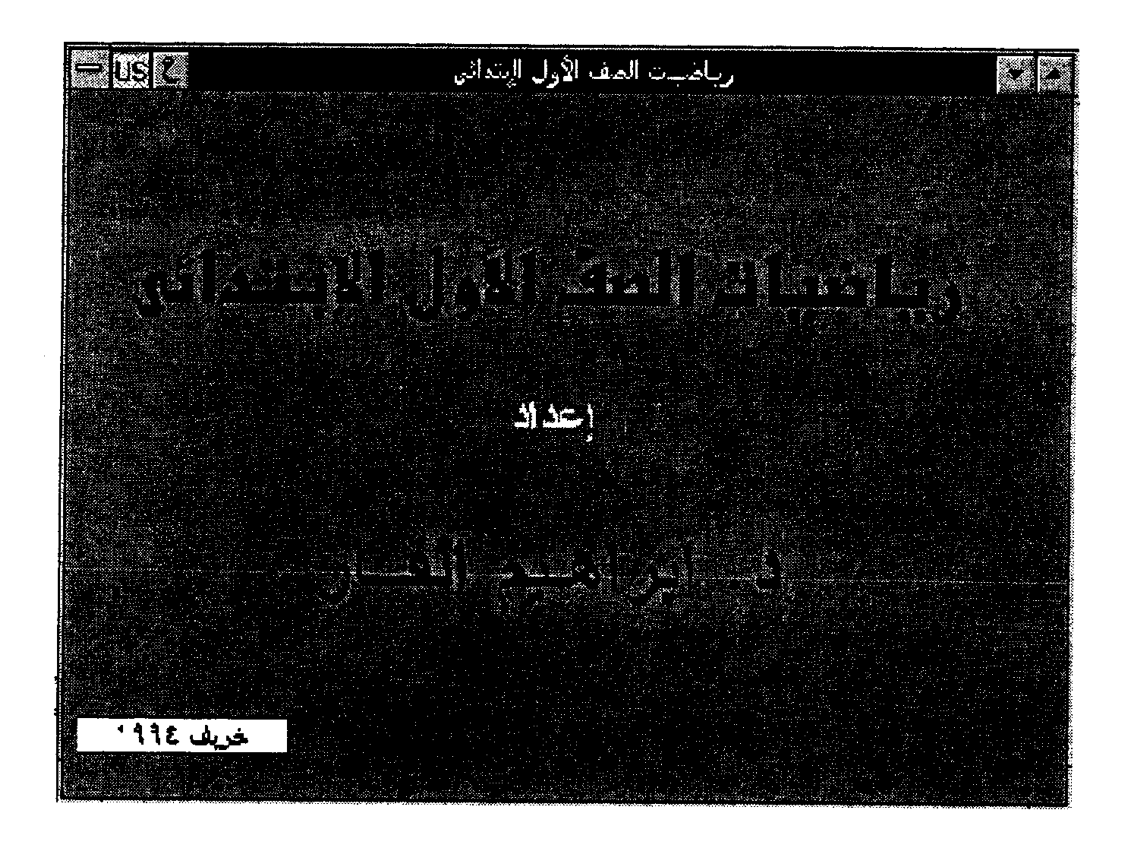


قم بتصميم شاشات المقدمة وشاشة المنتج كل على حدة ؛ وذلك بإختيارك للشاشة المطلوبة من صندوق المشروع ثم النقر على زر Veiw Form ، وسوف تنشط تلك الشاشة المختاره . ابدأ بالعمل على تصميمها والإنتهاء منها طبقا للسيناريو المحدد.

تدريبات وأمثلة لإعداد بعض الشاشات تصلح كشاشات مقدمة

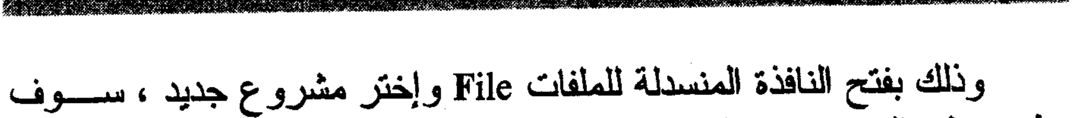
تدريب (١)

يتكون هذا التدريب من نافذة واحدة تحتوى على مجموعة من وحـــدات التحكم لإضافة النصوص Label ؛ أنظر شكل رقم (٩). وكما هو واضح من الشكل تعرض النافذة أربعة نصوص مختلفة.

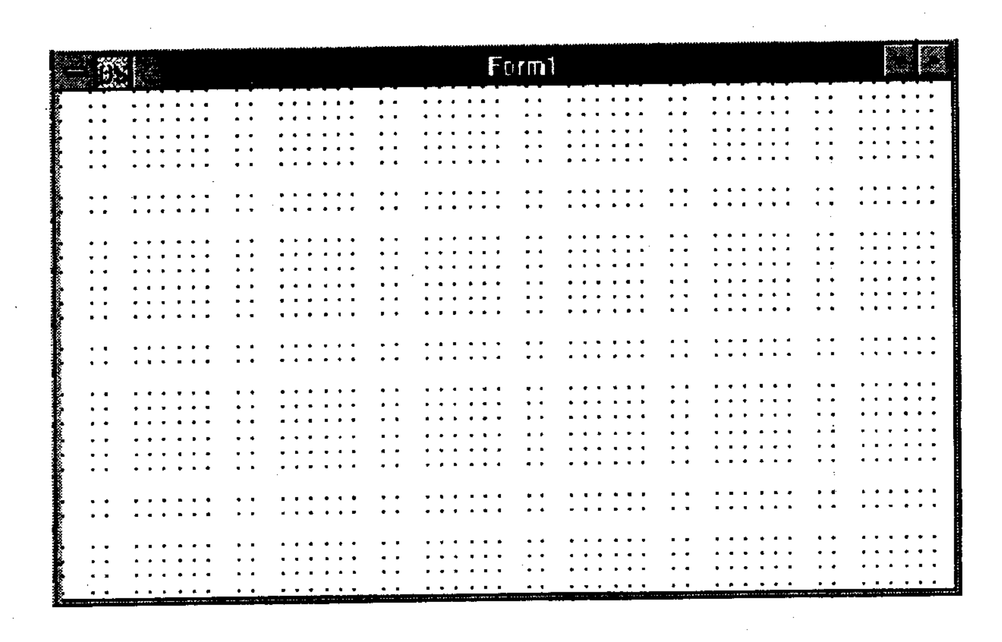


شكل رقم (٩)

شاشة مقدمة تعرض بعض النصوص



وذلك بفتح النافذة المنسدلة للملفات File وإختر مشروع جديد ، سـوف يظهر على الفور شاشة المسـتخدم User's Interface (النمـوذج) بالمسـمى المبدئي Form1



تحديد خصائص النموذج Form1

يمكن تحديد خصائص النموذج Form1 وذلك بإستخدام نافذة الخصـائص Properties Windows كما سبق لتصبح كماهى موضحة بجدول خصـائص Form1 التالى :

خصائص Form1		
Control	Property	Value
Form1		
	Captions	رياضيات الصف الأول الإبتدائي
	BackColor	&H00FF00FF& اللون البنفسجى
	Height	5550
	Left	1440
	Top	1100
	Width	7500

يمكن تحديد أبعاد النموذج بالفأرة كما هو متبع فى نظام وندوز

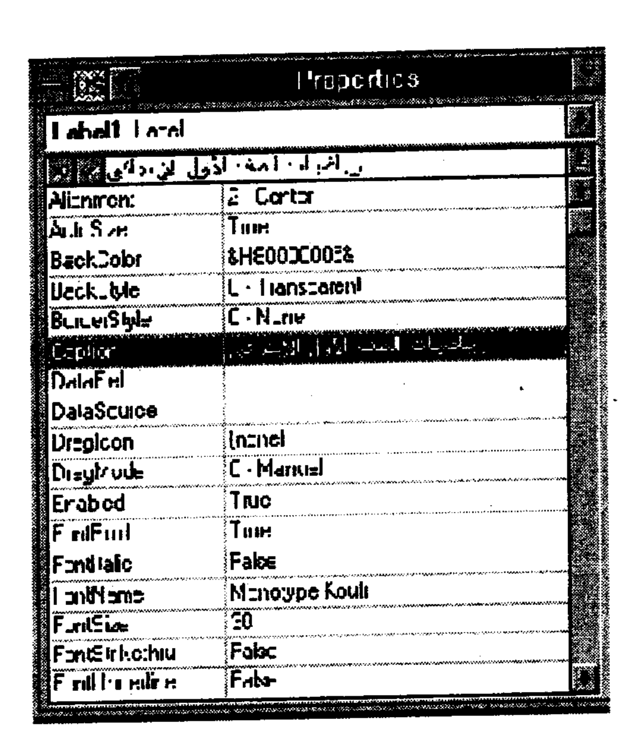
أداة إضافة النصوص Label

تعمل أداة إضافة النصوص على عرض نص بحيث لايسمح للمتعلم بتغييره ،بينما يمكن للمبرمج تغيرة فى مرحلة التصميم Design Time

إضافة صندوق النصوص Label Control

لإضافة صندوق النصوص بشاشة النموذج ، أنقر أداة النصوص سوف يظهر لك صندوق النصوص على شاشة النموذج بإسم مبدئى default name (Label1) ؛ إسحب الصندوق بالفأرة الى الموقع المناسب بالنموذج؛ علمه للشروع فى تحديد خصائصة.

يمكن تحديد خصائص Setting Properties صندوق النصوص Label1 ، وذلك بفتح النافذه المنسدلة Windows وإختيار Properties ، سوف تفتح لـــك نافذة الخصائص والموضحة بالشكل رقم (١٠) ؛ عدل الخصائص كمـــا هـــو موضح بجدول خصائص Label1 :



شكل رقم (١٠)

نافذة الخصائص

خصائص Label1		
Control	Property	Value
Label1		
	Alignment	2- Center لمركزة النص
	AutoSize	True
	BackColor	&H80000005&
	BackStyle	0- Transparent لإخفاء الصندوق
	Caption	رياضيات الصف الأول الإبتدائي
	FontBold	True
	FontName	MonoType Koufi الخط الكوفى
	FontSize	30
	ForeColor	&H00000000& لون الحرف

لاحظ أن أغلبية الخصائص تشتمل على إختيارين أو أكثر : يمكن تحديد الإختيار المطلوب إما بكتابته فى الجزء المخصص لذلك أو بفتح صندوق التحديدات الواقع أعلى نافذة الخصائص ؛ فمثلا :

خاصية Alignment والتى عن طريقها يمكن تحديد موقع النص داخل صندوق النصوص تتضمن الإختيارات التالية :

0 - Left Justify لمحازات النص جهة اليسار
1 - Right Justify لمحازات النص جهة اليمين
2 - Center لمركزة النص داخل صندوق النصوص

خاصية AutoSize والتى عن طريقها يمكن تحـديد حجـم صـــــــندوق النصوص أتوماتيكيا طبقا لطـــول النص تتضمن الإختيارات التالية :

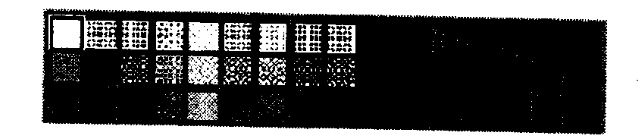
0 – True للتحديد الأتوماتيكى
1 – False لعدم التحديد الأتوماتيكى

خاصية BackStyle والتى عن طريقها يمكن التحكم فى ظهور صندوق النصوص حول النص من عدمه تتضمن الإختيارات التالية :

0 – Transparent لعدم إظهار صندوق النصوص
1 – Opeque لإظهار صندوق النصوص

لاحظ أنك قد لاتحتاج الى تعديل بقية الخواص حيث تترك كما هى بقيمها المبدئية Defualt ، وعليك تعديل الخواص التـــى تـؤدى الـــى تحسين عملك مرحليا فقط .

لاحظ أنه للسهولة عند الرغبة فى تحديد بعض الخواص كخـواص الألوان فإن برنامج البيسك المرئي يجلب لك بلتــة الألـــوان لتحديـــد إختيارك بسهولة : أنظر بلتة الألوان الموضحة بشكل (١١).

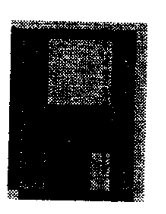


شكل رقم (١١)

بلتة الألوان



كرر الخطوات السابقة لتنفيذ بقية النصوص مع ملاحظة أن النص الرابع وهو (خريف ١٩٩٤) يظهر داخل صندوق النصوص .



خزن إنتاجك هذا بعد الإنتهاء منه على قرص مـــرن ، وهنـــا سوف يقوم نظام البيسك المرئى بتخزين المشروع على ملفيـــن حسب إختيارك الأول بإمتداد MAK. ، والثاني بإمتداد FRM. :

تدريب (٢)

يتكون هذا التدريب من نافذة واحدة تحتوى على مجموعة من وحـــدات التحكم لإضافة النصوص **Label** ؛ والصور **Picture** أنظر شكل رقـم (١٢). وكما هو واضح من الشكل تعرض النافذة خمسة نصوص مختلفة وصورة.



شكل رقم (١٢)

شاشة مقدمة

إبدأ مع مشروع جديد New Project

وذلك بفتح النافذة المنسدلة للملفات File وإختر مشروع جديد ، ســـوف يظهر على الفور شاشة النموذج Form1

حدد خصائص النموذج Form1

حدد خصائص النموذج Form1 عن طريق نافذة الخصائص Properties Windows كما سبق أو بالفأرة كما هو متبع فى نظام وندوز.

أداة إضافة الصور Picture

تعمل أداة إضافة الصور على عرض صورة علـــى النمــوذج بحيــث لايسمح للمتعلم بتغييرها ، بينما يمكن للمبرمج تغييرها فى مرحلة التصميــم ، هذا وتعمل أداة إضافة الصور هذه على إضافـــة صــورة ذات الإمتــدادت : **(.BMP , .WMF , .ICO , .DIB)**

إضافة صندوق الصور Picture Box

لإضـــافة صندوق الصـــور بشـــاشة النمـــوذج ، أنقـــر أداة الصـــــــــور يظهر لك صندوق الصور على شاشة النموذج بإسم مبدئى default name (Picture1) ؛ إسحب الصندوق بالفأرة الى الموقــع المناسب بالنموذج؛ علمه للشروع فى إضافة الصورة المطلوبة أليه وتحديـــد خصائصها.

تحديد الصورة Picture1 وخصائصها

يمكن تحديد الصورة Picture1 وخصائصـــها بفتــح النـــافذه المنســدلة Windows وإختيار Properties ، سوف تفتح لك نافذة الخصائص الموضحة بالشكل رقم (١٣) :

Properties	
Picture1 PictureBox	
(none)	
ForeColor	&H80000008&
Height	495
HelpContextID	0
Index	
Left	3120
LinkItem	
LinkMode	0 - None
LinkTimeout	50
LinkTopic	
MousePointer	0 - Default
Name	Picture1
Picture	(none)
ScaleHeight	465
ScaleLeft	0
ScaleMode	1 - Twip
ScaleTop	0
ScaleWidth	1185
TabIndex	0

شكل رقم (١٣)

نافذه تحديد الصورة وخصائصها

لاحظ أنه عندما تفتح نافذة تحديد الصورة وخصائصها تحدد خاصية صورة Picture أتوماتيكيا بالمحدد (none) ويظهر هذا المحدد بصندوق التحديديات .

لتحديد الصـــورة إضغط بالفأرة علــــى المربع ▦ الموجود على يمين صندوق التحديدات سوف يأخذك نظام البيسـك المرئي الى شاشة خاصة وهى الشاشة الموضحة بالشكل رقم (١٤) لتحميـــل الصورة المطلوبة من أى مصدر كان بشرط أن تكون ذات الإمتـــداد (.bmp) أو (.wmf) أو (.ico) أو (.dib)

شكل رقم (١٤)

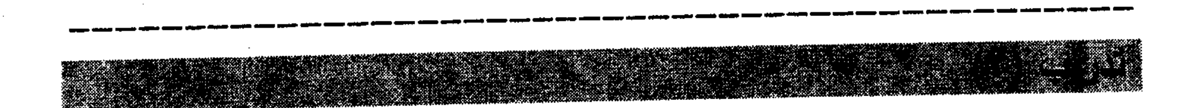
نافذه تحميل الصورة

بعد إضافة الصورة المختاره من المصدر الذى حددته وبالإمتداد السابق عليك أن تعدل الخاصـــــية AutoSize لتصـــــبح True ، وبعـــد ذلك حدد موقع الصورة عن طريق نافذة الخصائص بتحديد قيم الخصـــائص (Height , Left , Top , Width) أو عن طريق تحديد موقع الصورة بالفأرة .



أضف النصوص المطلوبة وذلك بإستخدام أداة عرض النصوص كما سبق.

خزن إنتاجك هذا بعد الإنتهاء منه على قرص مرن.



يتكون هذا التدريب من نافذة واحدة توضح إمكانيـــــة اضافـــة بعـض المؤثرات الخاصة التى توحى بالحركة مع تغيير الألوان والخطوط وحجمها : أنظر الشكلين أرقام (١٥) ، (١٦).

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

رياضيات الصف الأول

شكل رقم (١٥)

شاشة مقدمة توحى بالحركة



شكل رقم (١٦)

شاشة مقدمة توحى بحركة أكثر

وسوف تختلف طريقة أنشاء الشاشاتين الموضحتيـــن بالشـــكلين (١٥) ، (١٦) عن الطرق المستخدمة فى أنشاء الشاشات السابقة ، حيث سوف تعتمـــد على كتابة مايسمى بالكود (Code) البرنامج ، وذلك كالتالى .

إبدأ مشروع جديد New Project

وذلك بفتح النافذة المنسدلة للملفات File وإختر مشروع جديد ، ســـوف يظهر على الفور شاشة النموذج Form1

حدد خصائص النموذج Form1

حدد خصائص النموذج Form1 عن طريق نافذة الخصائص Properties كما سبق أو بالفأرة كما هو متبع فى نظام وندوز Windows.

```
Sub Form _Click ()
Dim  CX, I, J, K, L, M, Msg, Offset     ' تعيين أبعاد مجموعة متغيرات
Cls       ' لمسح ما بداخل النافذة
Width = 7200   ' يحدد عرض نافذة النموذج
Height = 5000   ' يحدد إرتفاع نافذة النموذج
BackColor = QBColor(1)      ' يحدد اللون البنفسجي للخط
For I=1 To 3        ' حلقة من ثلاث حالات
Select Case I
Case 1    ' الحالة الأولى
ForeColor = QBColor(5)    ' يحدد اللون البنفسجي للخط
K = 1 : L = 9 : M = 1
Msg = "الصف"    ' النص الذى يعرض على يسار النافذة
CX = 0  ' متغير يحدد لبمكان الذى يعرض فيه النص
Case 2    ' الحالة الثانية
K = 1 : L = 9 : M = 1

Msg = " رياضيات "   ' النص الذى يعرض على يمين النافذة
      CX = ScaleWidth  ' عرض النافذة المحدد فى جدول الخصائص

              Case 3    ' الحالة الثالثة
                ForeColor = QBColor(15)  ' يحدد اللون الأبيض للخط
                  K = 9 : L = 1 : M = -1
                  Msg = " رياضيات الصف الأول "
                  CX = ScaleWidth / 2  ' يعرض النص فى منتصف النافذة

              End Select

      For J = K To L Step M
```

```
Select Case J  ' حلقة مكونة من ثلاثة متغيرات تنادى على القيم من
' الحالات الثلاث السابقة لتغيير مقاس الخط تسع مرات

        Case 1 : Fontsize = 8
        Case 2 : Fontsize = 10
        Case 3 : Fontsize = 12
        Case 4 : Fontsize = 14
        Case 5 : Fontsize = 18
        Case 6 : Fontsize = 20
        Case 7 : Fontsize = 24
        Case 8 : Fontsize = 36
        Case 9 : Fontsize = 48

    End Select

     If I = 1 Then
         Offset = 0   ' النص على اليمين

     ElseIf I = 2 Then
         Offset = TextWidth(Msg)  ' النص على اليسار
     Else

         Offset = TextWidth(Msg) / 2  ' النص فى المنتصف
     End  If

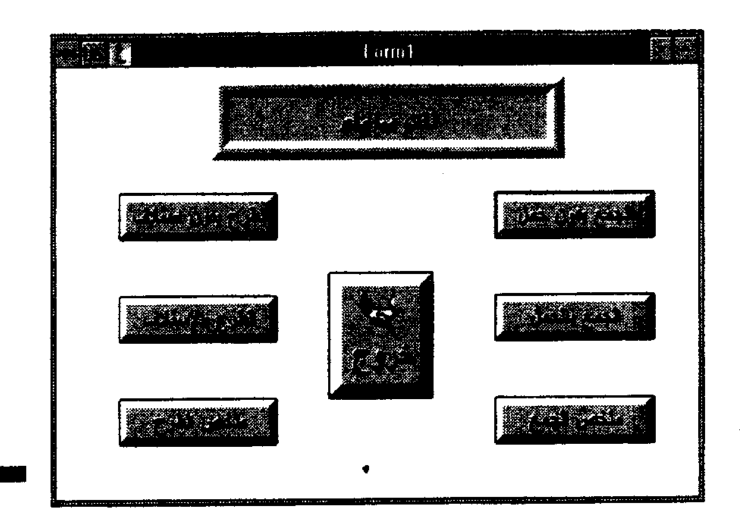
  CurrentX = CX - Offset   ' يعود موقع الطباعة الى ماكان عليه
   Print Msg   ' يطبع النص

     Next J

  CurrentY = 0   ' يعود موقع الطباعة الى أعلى النافذة

Next I

End Sub
```



الفصل السابع

تصميم وإنتاج شاشة

القائمة الرئيسية والقوائم الفرعية

تمهيد

يحتاج التطبيق البسيط عادة إلى عدد محدود من أدواة التحكم (صنـــدوق حوار، زر إختيار ، ...) ، بينما تحتاج بعض التطبيقات الي العديد من تلـــك الأدوات ، وبالتالي نحتاج الى عدد من الشاشات ، ويصبح الأمر اكثر صعوبة للمستخدم عند الشروع فى التعامل مع تلك الشاشات . ولحل هذه المشكلة فانه لامناص من إستخدام مايسمى بالقوائم menues : ويمكن للمبرمـــج يســتخدم اكثر من تكنيك لتصميم هذه القائمه فى صورة واجهة متقنة وسهلة الإســتخدام للمتعلم، وعادة فى البرمجية التعليمية مايستخدم المبرمج احد التكنيكات التالية، طبقا لطبيعة الدرس وخصائص المتعلم :

- طريقة القوئم المنسدلة (شريط القوائم Menu Bar)
- طريقة القائمه الحره Free Menu

طريقة القوائم المنسدلة (شريط القوائم Menu Bar)

وهو مايعرف فى نظام ويندوز بالقوائم المنســـدلة *drop-down menues* حيث يستطيع المبرمج أن يصمم شريط قوائم في مشـــروعة ويكـــون متـــاح للمتعلم في وقت التشغيل . وينبغى اعداد بناءات منطقية لتلك القوائم لتكـــون ديناميكية وتستجيب لتشعيب المتعلم أو لظروف التشغيل.

نافذة تصميم شريط القوائم Menu Design Window

تتكون نافذة تصميم القائمة والموضحه بالشكل رقم (١٧) مـــن جزئيـــن اساسيين **الجزء الأول** : وهو الخاص بالتحديدات والذى يشتمل على بعـــض الأدوات فى صـــورة الصناديق والاختيـــارات التاليـــة : Name ، Caption ، Index ، Shortcut ، WindowList ، HelpContextID ، Chected ، Enabled ، Visible

والجزء الثانى : وهو صندوق قائمة المخطط Menu Outline الذى يظهر فيـه المخطط العام للقوائم، اضافة الى عدد من الأزرة التى تساعد المبرمج علـــى تحديد مستويات والقوائم وما تحتويه من اختيارات.

Menu Design Window

Caption: OK

Name: Cancel

Index: Shortcut: (none)

WindowList HelpContextID: 0

Checked Enabled Visible

Next Insert Delete

شكل رقم (١٧)

نافذة تصميم شريط القوائم

الأدوات المتاحة بالجزء الأول من نافذة تصميم القوائم

صندوق تحديد العنوان Caption

وهو الذى يمكن المبرمج من تحديد عنوان لكل قائمة قيد الإنشاء ، وهو ذات العنوان الذى سوف يظهر للمستخدم فى مرحلة التنفيذ كعنوان للقائمة المنسدلة ، مثل عنوان القائمة المنسدلة File مثلا بنظام ويندوز.

صندوق تحديد الإسم Name

وهو الذى يمكن المبرمج من تحديد اسم لكل قائمة تم عنونتها عن طريق الخاصية السابقة Caption ، هذا وسوف يستخدم هذا الإسم كإسم متغير variable name عند كتابة البرنامج للتحكم فى تدفق الشاشات الخاصه بكل اختيار من اختيارات القائمة ، ويفضل ان يتكون من مقطعين لسهولة المتابعة : مقطع دال عن عنوان القائمة الرئيسية mnu مثلا ، وآخر دال عن عنوان الاختيار كا Option1 ، فتبدوا اسماء الاختيارات على سبيل المثال كالتالى : mnuOption1 ، mnuOption2 ، .....

صندوق تحديد الفهرس Index

يمكن عن طريقه تحديد مواقع اسماء متغيرات الاختيارات mnuOptions حالة ما يتم التعبير عنها كمتغيرات مذيلة (متغيرات ذات ابعاد array) ، وذلك لأهميته اذا ماكان التطبيق يضيف instance جديد فى قائمة التحكم خلال التنفيذ، ويترك الصندوق فارغا حالة عدم التحديد

صندوق تحديد مفاتيح اختزال الخطوات Shortcut

يمكن عن طريقه تحديد مفتاح ما من مفاتيح الوظائف Function Key أو اى توليفة من مفتاحين مثل (Ctrl+M) مثلا والتى تعرف بمفاتيح اختزال الخطوات . وبالتالى يمكن للمستخدم استخدامها لتنشيط اختيار ما او وظيفة ما من داخل قائمة منسدلة مباشرة بدلا من فتح تلك القائمة ثم اختيار تلك الوظيفة (مثل استخدام Ctrl+S للحفظ مباشرة فى نظام ويندوز بدلا من اختيار القائمة المنسدلة File ثم اختيار وظيفة Save)، وترك الصندوق كما هو (none) حالة عدم التحديد ، أما فى حالة فتحة لاختيار للتحديد المطلوب فسنجد انه يحتوى على كافة المفاتيح والتوليفات المتاحة.

(none)
(none)
Ctrl+A
Ctrl+B
Ctrl+C
Ctrl+D
Ctrl+E
Ctrl+F
Ctrl+G
Ctrl+H
Ctrl+I
Ctrl+J
Ctrl+K

خاصية قائمة النوافذ WindowList

باختيارها يتمكن المبرمج من تصميم قائمة من النوافذ السابقة الفتح من خلال نافذة الوثائق المتعددة MDI ، والوضع المبدئى Defualt لهذه الخاصية هو عدم الإختيار.

صندوق تخليق المساعدة بالإحساس HelpContexID

باختيار هذه الخاصية يتمكن المبرمج من تصميم نظام المساعدة بالإحساس *Context-sensitive Helpsystem* ، والوضع المبدئى Defualt لهذه الخاصية هو القيمة صفر (0)

خاصية وضع علامة على القائمة الفرعية سابقة الإختيار Checked

باختيارها تعمل على اظهار علامة check mark على القائمة الفرعية السابق تحديدها تحديدا مبدئيا Defualt عند التعامل مع القائمة الاساسية ، والوضع المبدئى Defualt لهذه الخاصية هو عدم الاختيار.

خاصية عدم الإمكان Enabled

باختيار هذه الخاصية تعمل على اظهار الاختيارات التى يجب ان لا تستخدم فى موقف ما بلون رصاصى فاتح لتبلغ المستخدم بعدم امكانية استخدام تلك الإختيارات فى هذا الموقف، والوضع المبدئى Defualt لهذه الخاصية هو الإختيار .

خاصية الإختفاء Visible

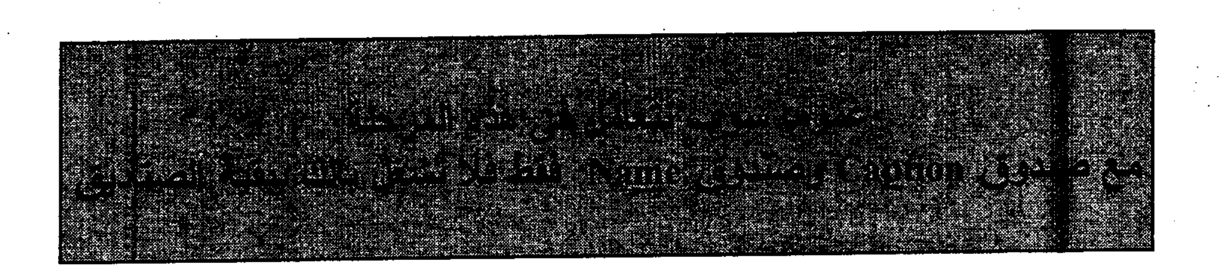
تعتبر هذه الخاصية ذات اهمية فى بعض التطبيقات الخاصة ، وهى تختلف عن خاصية عدم الإمكان Enabled حيث انه بعدم اختيارها تعمل على اختفاء القوائم المنسدلة التى تم تصميمها بالكامل. وبالتالى فان والوضع المبدئى Defualt لهذه الخاصية هو الإختيار .

زر الموافقة OK *butten*

يعمل هذا الزر عند الضغط عليه على اغلاق نافذة تصميم القوائم مع تسجيل كافة ماتم تصميمة ، ويتم الضغط عليه بالطبع بعد الإنتهاء من تصميم كافة القوائم المطلوبة أو جزء منها.

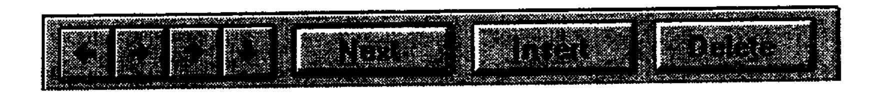
زر الإلغاء Cancel *button*

يعمل هذا الزر عند الضغط عليه على اغلاق نافذة تصميم القوائم مع اهمال كافة ماتم تعديلة.



الجزء الثاني : قائمة المخطط Menu Outline

وهو عبارة عن صندوق يظهر فيه المخطط العام للقوائم المعنية بالتصميم وما تحتويه من اختيارات ، ويوجد أعلا هذا الصندوق عدد من الأزرة button



التى تساعد المبرمج فى تحديد مستويات القوائم وماتحتويه من اختيارات.

ازرة الأسهم	تمكن المبرمج من تحديد وتغيير مستويات الاختيارات الفرعية داخل القوائم. هذا ويمكنك تخليق حتى اربعة مستويات فرعية لكل قائمة رئيسية.
زر التتالى Next	تمكن المبرمج من تحريك المؤشرة الى اسفل داخل قائمة المخطط.
زر الإقحام Insert	وهو الذى يمكن المبرمج من حشر أو اقحام اختيار جديد أو قائمة جديدة داخل سطور المخطط.
زر الالغاء Delete	وهو الذى يمكن المبرمج من حذف اختيار ما من أو قائمة من داخل المخطط.



يمكنك التعديل : بالحذف او الاضافة أو التنسيق داخل صندوق المخطط متى استلزم الأمر ذلك بالعودة الى نافذة تصميم القوائم.

خطوات تصميم شريط قوائم Menu Bar

ولتصميم شريط القوائم المطلوب يجب إتباع الخطوات التالية :

١- اختر النافذة التي تريد تصميم شريط قوائم عليها من نافذة المشروع لتكن **Form4** والضغط على ***View Form*** .

٢- من القائمة المنسدلة **Window** اختر ***Menu Design*** سيظهر صندوق الحوار الموضح بالشكل رقم (١٧)

ملاحظة ١

بالإضافة الى الطريقة السابقة ؛ يمكنك فتح نافذة تصميم قائمة Menu design Window بإتباع أى من الطريقتين التاليتين :

- النقرعلى زر تصميم القائمة من شريط الأدوات
- الضغط على مفتاحي Ctrl + M

ملاحظة ٢

اذا كنت قد استخدمت نظام النافذه المتعددة الوثائق Multiple Document Interface (MDI) ، تأكد من انك قد اخترت خاصية *activate* للنماذج التى سوف تتعامل مع القائمة قبل الشروع فى فتح نافذة تصميم القائمة .

٣- أكتب عنوان القائمة كما تريده أن يعرض في صندوق **Caption**

٤- أكتب اسم القائمة باسم *Variable Name* كما تريده أن يعرض في صندوق **Name** لتستخدمه عند كتابة البناء المنطقي Code

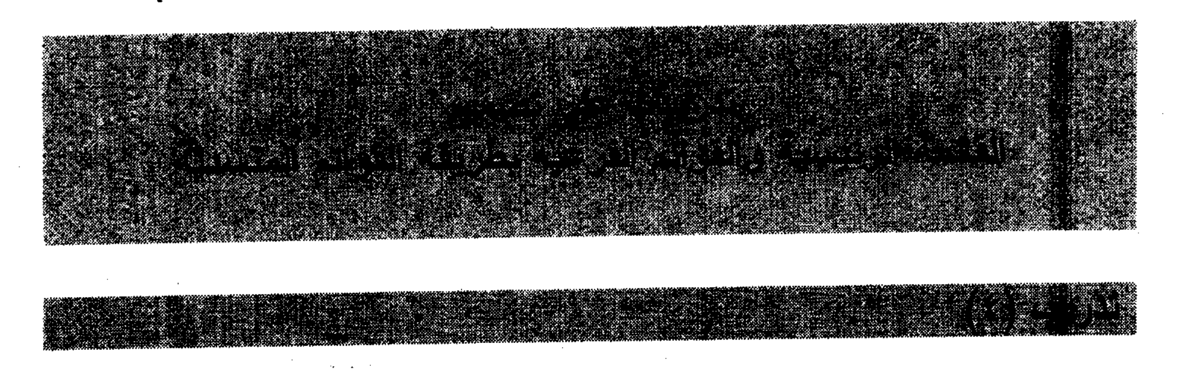
٥- استخدم الأزره المتاحة لتنسيق القوائم كالتالي :

- زرا السهم المتجه إلى اليمين والسهم المتجه إلى اليسار لتحديد المستويات الرأسية للقوائم (أربعة مستويات كحد أقصى).
- زرا السهم المتجه إلى أعلى والسهم المتجه إلى أسفل لتحديد الترتيب الأفقي للقوائم .
- زر Next لتتحرك إلى السطر التالي .

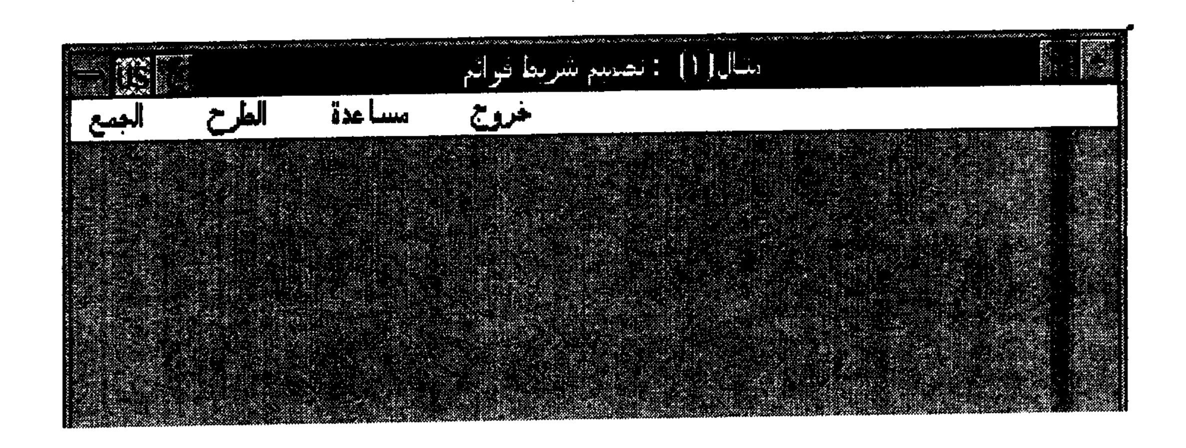
- زر Insert لإدخال (حشر) سطر خالي .
- زر Delete لإلغاء السطر معين .

٦- كرر الخطوات : ٣ ، ٤ ، ٥ لإضافة عنصر جديد للقائمة.

٧- اضغط OK عند الإنتهاء من تصميم القائمة ؛ لتسجيل القائمة على النموذج المستهدف ، والخروج من مرحلة تصميم شريط القوائم ، وبالطبع سوف يعرض شريط القائمة الذي تم تصميمه وقت التنفيذ ، ولكن دون تصيد الأحداث *Event Trapping* إلى آن تكتب البناءات المنطقية . يمكنك الضغط على Cancel للخروج من تصميم القائمة دون اتمامها ، والعودة لتصميمها من جديد.



صمم الشاشة الرئيسية التالية والموضحة بالشكل رقم (١٨)



شكل (١٨)

شاشة رئيسية

موضحا عليها شريط قوائم منسدله

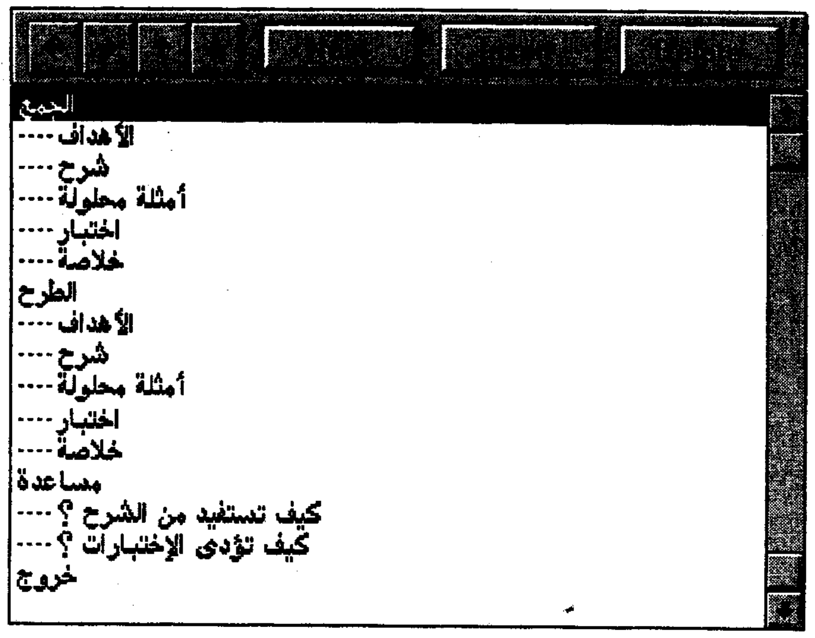
بحيث تبدو القوائم المنسدلة عند فتحها كالتالى :

خروج

مساعدة
- كيف تستفيد من الشرح ؟
- كيف تؤدى الإختبارات ؟

الطرح
- الأهداف
- شرح
- أمثلة محلولة
- اختبار
- خلاصة

الجمع
- الأهداف
- شرح
- أمثلة محلولة
- اختبار
- خلاصة



صندوق المخطط كما يبدو فى هذا التدريب

قائمة مواصفات شريط القوائم

Caption		Name
القائمة	الإختيارات	
الجمع		Addition
	الأهداف	AddObject
	شرح	AddPresentation
	امثلة محلولة	AddExamples
	اختبار	AddTest
	خلاصة	AddSummary
الطرح		Subtraction
	الأهداف	SubObject
	شرح	SubPresentation
	امثلة محلولة	SubExamples
	اختبار	SubTest
	خلاصة	SubSummary
مساعدة		Help
	كيف تستفيد من الشرح ؟	HelpPresentation
	كيف نؤدى الإختبارات ؟	HelpTest
خروج		Quit

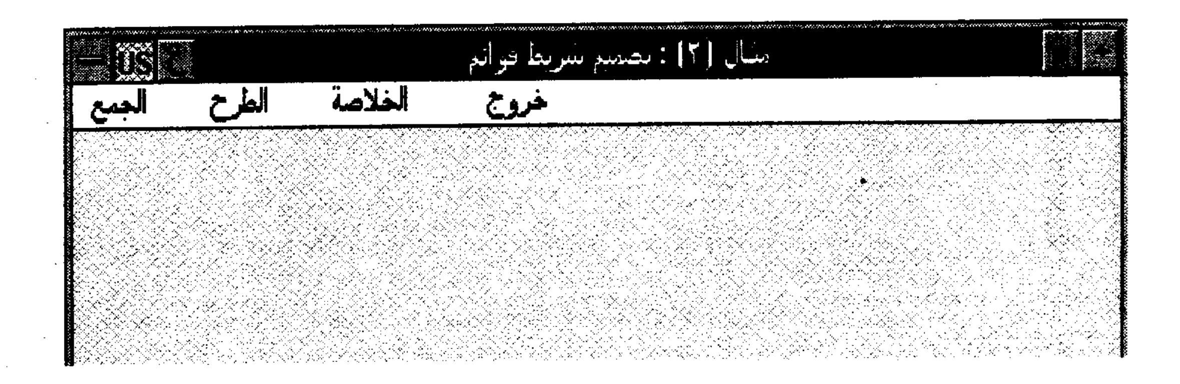
ملاحظة ١
يفضل كتابة العنوان Caption باللغة العربية ، بينما يفضل كتابة الإسم Name باللغة الإنجليزية .

ملاحظة ٢
للإنتقال من صندوق العنوان Caption الـــى صنـــدوق الإســـم Name ؛ استخدم الفأرة أو مفتاح Tab

ملاحظة ٣
بعد الإنتهاء من كتابة العنـــوان والإســـم للقائمـــة أو الإختيـــار الفرعى ؛ اضغط على زر Next لتنتقـــل الـــى ســـطر جديـــد بصندوق المخطط لتتمكن من كتابة اسم وعنوان جديدين.

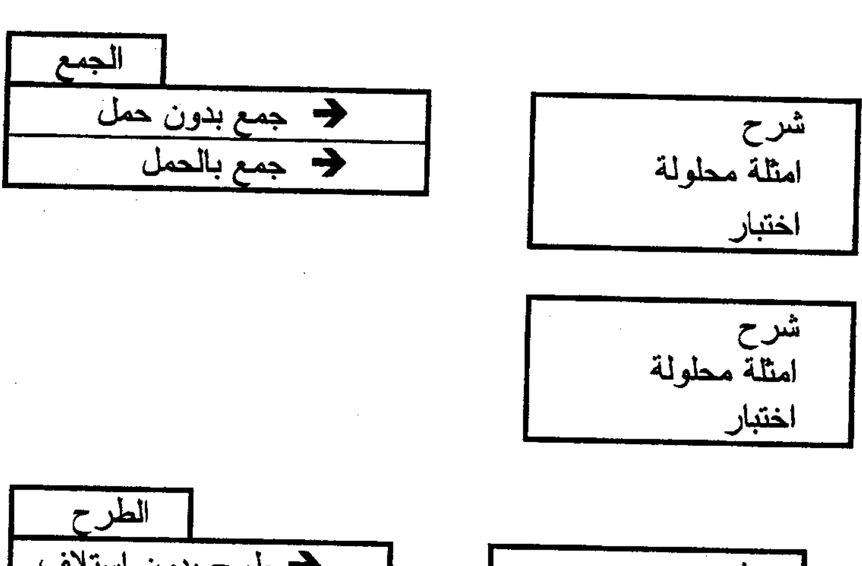
التدريب (٥)

صمم الشاشة الرئيسية التالية والموضحة بالشكل رقم (١٩)



شكل (١٩)
شاشة رئيسية
موضحا عليـــها شـــريط قوائـــم
منسدلة

بحيث تبدو القوائم المنسدلة عند فتحها كالتالى :



الطرح

طرح بدون استلاف

طرح بالإستلاف

شرح

امثلة محلولة

اختبار

شرح

امثلة محلولة

اختبار

الخلاصة

ماذا تعلمت فى الجمع ؟

ماذا تعلمت فى الطرح ؟

الأسماء والعناوين كما تبدو فى صناديق Caption & Name

Caption			Name
القائمة الرئيسية	الإختيارات	المستوى الأول للاختيارت	
الجمع			Addition
	جمع بدون حمل		AddNoGrouping
		شرح	AddNGPresentation
		امثلة محلولة	AddNGExamples
		اختبار	AddNGTest
	جمع بالحمل		AddWithGrouping
		شرح	addWGPresentation
		امثلة محلولة	addWGExamples
		اختبار	addWGTest
الطرح			Subtraction
	طرح بدون استلاف		subNoBarow
		شرح	subNBPresentation
		امثلة محلولة	subNBExamples
		اختبار	subNBTest
	طرح بالاستلاف		subWithBarow
		شرح	subWBPresentation
		امثلة محلولة	subWBExamples
		اختبار	subWBTest
الخلاصة			Conclusion
	ماذا تعلمت فى الجمع ؟		conAddition
	ماذا تعلمت فى الطرح ؟		conSubtraction
خروج			Quit



صمم الشاشة الرئيسية التالية والموضحة بالشكل رقم (٢٠)

مثال (٣) على تصميم شريط القوائم

ملف   تحرير   التحديدات   حول

شكل (٢٠)

شاشة رئيسية
موضحا عليها شريط قوائم منسدلة

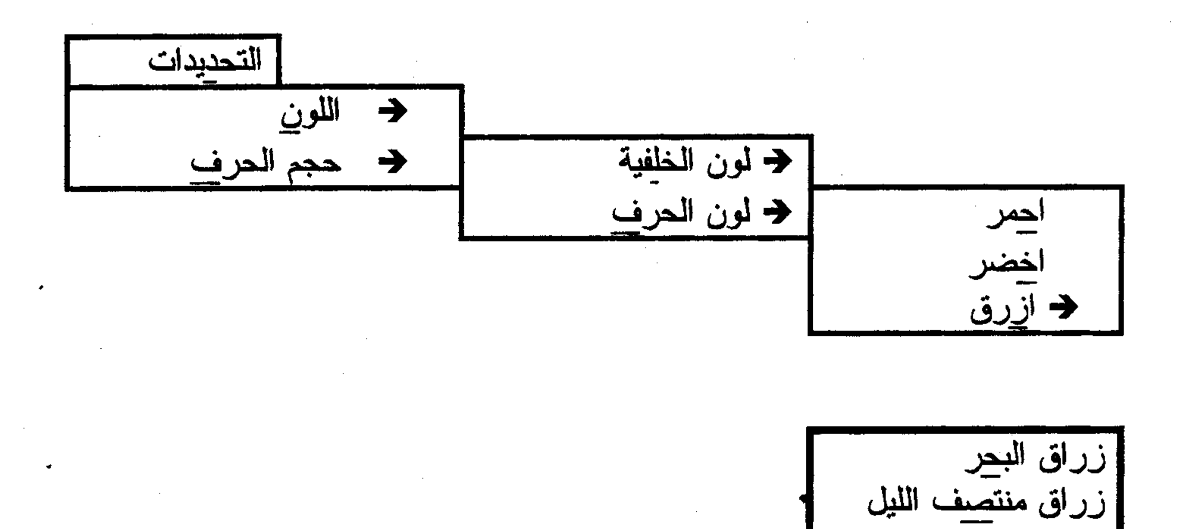
بحيث تبدو القوائم المنسدلة عند فتحها كالتالى :

ملف

- جديد
- فتح ...
- حفظ باسم ...
- غلق
- خروج

تحرير

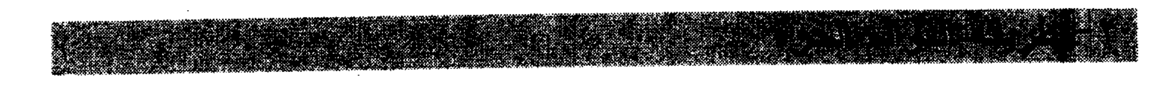
قص	Ctrl + X
نسخ	Ctrl + C
لصق	Ctrl + V



←	→	↑	↓	Next	Insert	Delete

&ملف
----ج&ديد
----ف&تح ...
----حفظ با&سم ...
----&غلق
----‐
----خ&روج
----‐
ت&حرير
----&قص Ctrl+X
----&نسخ Ctrl+C
----ل&صق Ctrl+V
التح&ديدات
----&اللون
--------لون الخ&لفية
------------اح&مر
------------اخ&ضر
------------از&رق
--------لون الحر&ف
------------ا&حمر
------------ا&خضر
------------از&رق
----------------ازرق ف&اتح
----------------ازرق غ&امق
--------------------زراق البح&ر
--------------------زراق منتص&ف الليل
----&حجم الحرف
--------12
--------24
&حول

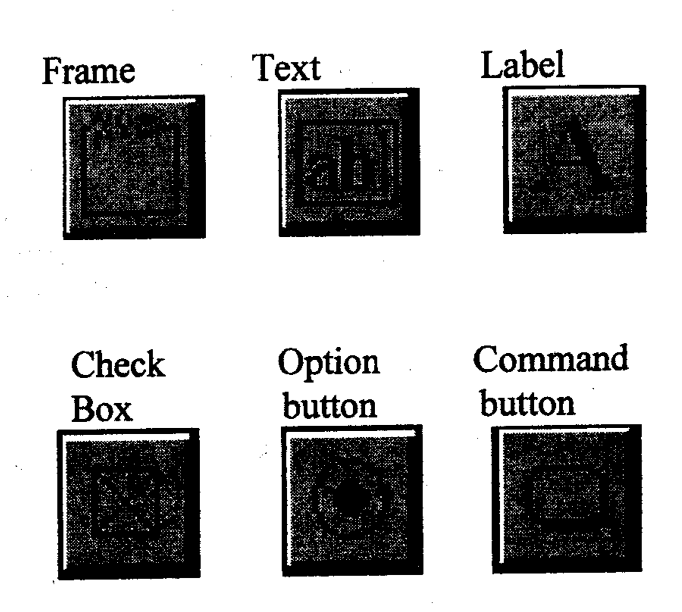
صندوق المخطط
كما يبدو فى هذا التدريب



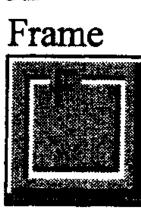
وهذه الطريقة قد تناسب الفئات العمرية الأولى من المتعلمين (طلاب المرحلة الإبتدائية مثلا) ؛ والذين لم يألفوا قوائم وندوز المنسدلة بعد ، وعادة ماتظهر هذه القائمة من خلال شاشة واحدة ، حيث يحتاج المبرمج وكاتب السيناريو أو الاثنين معا الى بعض الجهد والوقت لإخراج هذه الشاشة بشكل جيد .

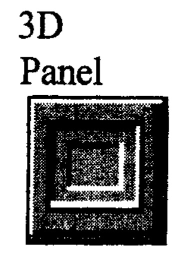
ومن خلال هذه الشاشة تظهر جميع اختيارات الدرس أو فقراته في صورة أزرة حره على شاشة القائمة ، وعلى المتعلم ان يختار مايريد من الفقرات بالنقر على زر الفقرة المختارة ؛ فيقوده الحاسوب الى تلك الفقرة التى يرغب فى التعامل معها (شرح أو تمرين أو تأدية اختبار أو مراجعة ... إلخ).

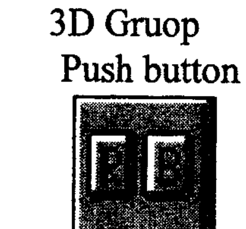
وهناك العديد من ادوات التحكم المتوافره فى بيئة بيسك المرئى التى تساعد المصمم او المبرمج فى تصميم شاشات القائمة بهذه الطريقة : منها وحدات التحكم العادية مثل :

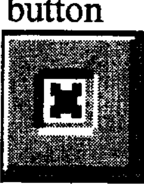


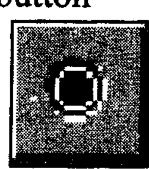
ومنها وحدات التحكم ثلاثية الأبعاد والتى تتميز عن الأولى بأنـــه عنـــد استخدامها تعطى للمشروع رؤية اعمق ومظهر أكثر جاذبيـــة مـــن وحـــدات التحكم العادية مثل :

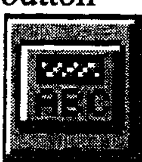
3D Frame


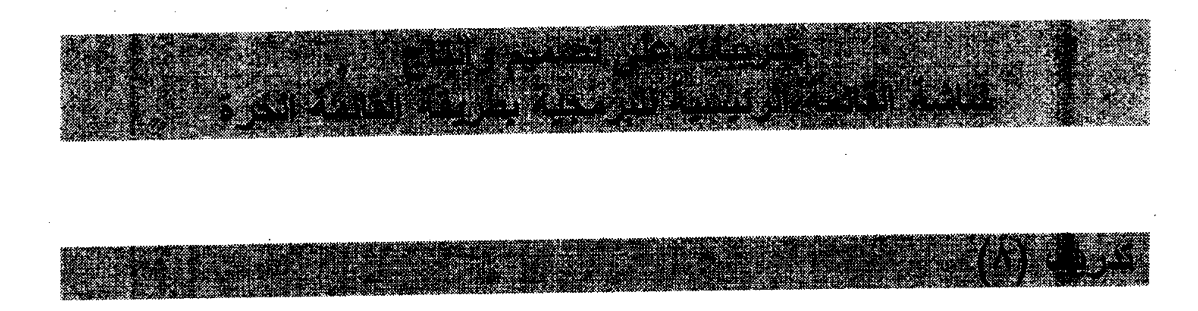
3D Panel


3D Gruop Push button


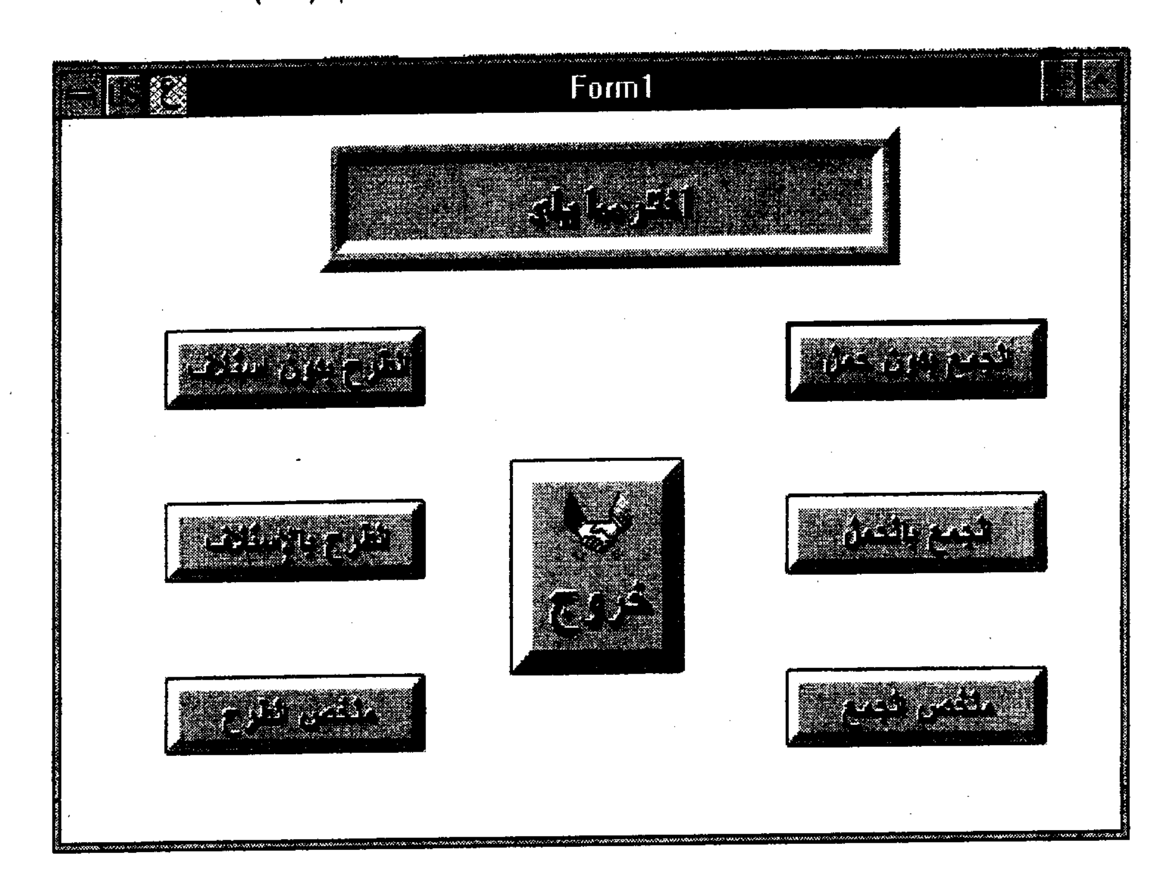
3D Check button


3D Option button


3D Command button


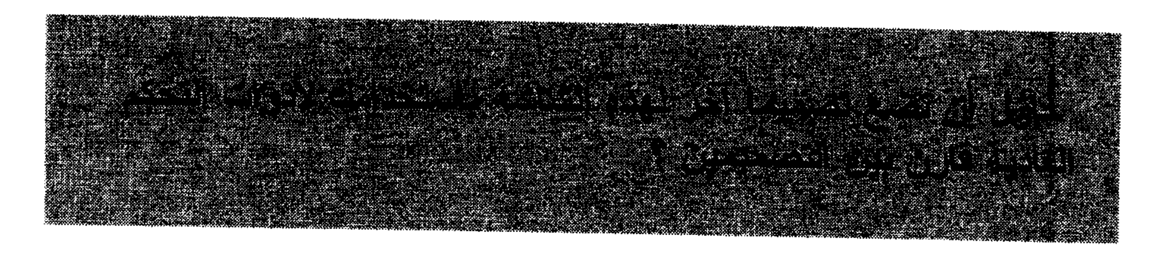


صمم الشاشة الرئيسية التالية والموضحة بالشكل رقم (٢١)



شكل (٢١)

شاشة رئيسية موضحا
عليها ازرة الفقرات كقائمة حرة



كما هو واضح بالشكل رقم (٢١) تشتمل شاشة القائمة على سبعة أزرة أوامر ثلاثية الأبعاد 3D Command button وبنل 3D panel علية جملة " **اختر مما يلى** " كلاصق ، ولتصميم هذه الشاشة اتبع الخطوات التالية :

١ – حدد الشاشة الرئيسية – من بين شاشات المشروع – التى ترغب طبقا للسيناريو ان تكون شاشة القائمة.



٢ – انقر اداة الأوامر الثلاثية الأبعاد 3D Command button بالفارة سوف يظهر على النموذج متوازى مستطيلات (صندوق ثلاثى الأبعاد) بالاسم المبدئى Command3D1 : اسحبه بالفأرة الى الموقع المحدد بالنموذج كأول زر الى جهة اليمين من اعلى (زر اختيار الجمع بدون حمل) .

٣ – اكتب عنوان الإختيار " الجمع بدون حمل" عن طريق خاصية Caption من خلال نافذة الخصائص.

عند هذه النقطة سوف يظهر صندوق الإختيار كمستطيل مستوى ، ويمكننا اظهار التجسيم باستخدام الخاصية :

BevelWidth : والتى عن طريقها يمكن التحكم فى درجة التجسيم وذلك من خلال مقياس يشتمل على القيم من (0 - 10)



لاحظ ان أزرة الأوامر الثلاثية الأبعاد والعادية مصممة لتظهر في افضل صورة لها باللون الرصاصي ، وهنا يمكنك تغيير لون خلفية النموذج لتصبح متناسقة مع لون الأزرة .

يمكنك التحكم فى لون الكتابة المحددة
لوظيفة الزر عن طريق الخاصية *ForeColor*

يمكنك التحكم فى درجة تظليل الخط المستخدم فى كتابة عنوان وظيفة الزر Caption وذلك عن طريق الخاصية :

Font3D : والتى عن طريقها يمكن التحكم فى درجة تظليل الخط المستخدم فى كتابة عنوان وظيفة الزر ، والذى يظهر كخط مجسم والتى تشتمل بدورها على الإختيارات التالية :

- 0 دون تظليل
- 1 بروز الحرف للخارج مع تظليل خفيف
- 2 بروز الحرف للخارج مع تظليل ثقيل
- 3 ضغط الحرف للداخل مع تظليل خفيف
- 4 ضغط الحرف للداخل مع تظليل ثقيل

يمكنك اضافة بعض التحسينات التى تساهم فى تحديد افضل للحدود الخارجية المحددة للأزرة ، وذلك عن طريق ان تكونا الخاصيتين التاليتين كما هو موضح :

Outline	False	وهذا يعنى عدم تحديد الأزرة من الخارج
RoundCorners	True	وهذا يعنى أن تكون اركان الأزرة مدورة

ملاحظة

يفضل بالطبع كتابة العنوان Caption باللغة العربية ، بينما يفضل كتابة الإسم Name باللغة الإنجليزية .

كرر الخطوات السابقة لتصميم الأزرة الست الباقية وهى الخاصة بالإختيارات: (الجمع بالحمل ، ملخص الجمع ، الطرح بدون استلاف ، الطرح بالإستلاف ، ملخص الطرح، خروج)

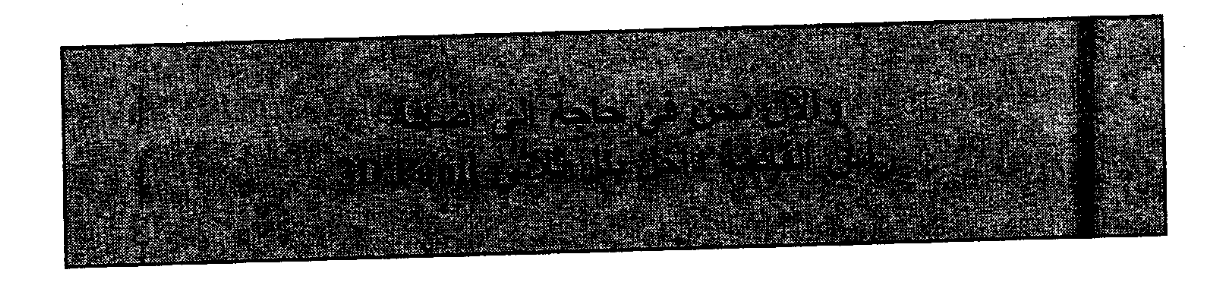
لاحظ ان الزر السابع وهو زر الخروج : اكبر حجما واكثر بروزا للخارج وعليه رسمة تعبيرية Icon لجذب انتباه المتعلم .

لاحظ ان اسم الزر سوف نستخدمه فى الكود كإسم متغير variable name وبالتالى فيفضل ان يحدد بطريقة تسهل علينا المتابعة عند كتابة الكود، وعلية فإنه – كما سبق الإشارة اليه عند تحديد اسماء اختيارات القائمة المنسدلة – نقترح ان تكون تلك الأسماء دالة كما هى موضحة بالجدول التالي :

اسماء اختيارات القائمة والتى سوف تستخدم كأسماء متغيرات

Caption	Name	
	Dafualt Name	Our Name
الجمع بدون حمل	Command3D1	AddNGrouping
الجمع بالحمل	Command3D2	AddWGrouping
ملخص الجمع	Command3D3	AddSummary
الطرح بدون استلاف	Command3D4	SubNBarow
الطرح بالاستلاف	Command3D5	SubWBarow
ملخص الطرح	Command3D6	SubSummary
خروج	Command3D7	Quit

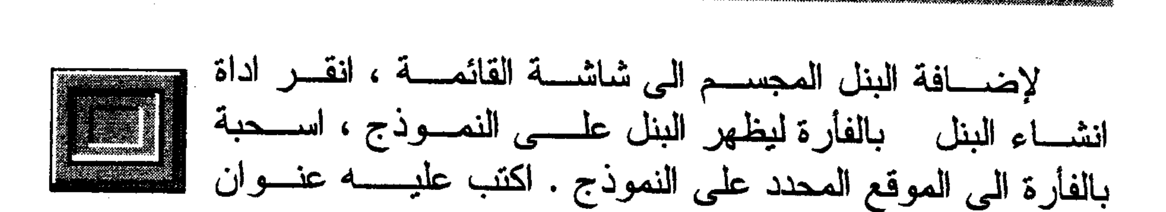
لإضافة الصورة التعبيرية Icon للزر السابع (زر خروج) : من صندوق الخصائص الخاص بالزر اختر خاصية Picture ، ومن خلالها اجلب الصورة المطلوبة والتى ستجدها فى الفهرس : C:\VB\COMM\HANDSHAK



[illegible] 3D Panil [illegible]

يشبه صندوق البنل الثلاثي الأبعاد (البنل المجسم 3D Panil) اللاصق label أو الإطار (الإطار العادى Frame أو الإطار المجسم 3D Frame) من ناحية الوظيفة ، ولكنه يتميز بمظهره الجذاب الذى يعمل على جذب انتباه المتعلم ، وبالتالى فهو يستخدم كإطار لتجميع بعض أدوات التحكم أو كلاصق ذو شكل متميز.

إضافة البنل المجسم كلاصق لشاشة القائمة



لإضافة البنل المجسم الى شاشة القائمة ، انقر اداة انشاء البنل بالفأرة ليظهر البنل على النموذج ، اسحبة بالفأرة الى الموقع المحدد على النموذج . اكتب عليه عنوان رأس القائمة " **اختر مما يلي** "

يتميز البنل المجسم بإحتوائه على بعض الخواص التى تساعد المصمم على تصميم وإخراج بنل ذو مظهر جذاب كالخواص التالية :

> ✍ خاصية *BevelWidth* والتى عن طريقها يمكن التحكم فى درجة انحدار حواف البنل beveled edges من الداخل والخارج ، حيث يمكن تحديد درجة الإنحدار من خلال مقياس يشتمل على القيم من (0 - 30).

✍ خاصـــية *BorderWidth* والتى عن طريقها يمكن التحكم فى درجة عمق البنل ، حيث يمكن تحديد درجة العمــق من خلال مقياس يشـــــــتمل على القيم من (30 - 0).

✍ خاصـــية *BevelInner* التى عن طريقها يمكن ضبط شكل الإنحدار الداخلي لحواف البنل ، والتى تشـــــتمل بدورها على الإختيارات التالية :

0 - None الخاصية لاتعمل
1 - Inset للداخل
2 - Raised للخارج

✍ خاصـــــــية *BevelOuter* التى عن طــريقها يمكن ضــبط شكل الإنحدار الخارجي لحواف البنل ، والتى تشتمل بدورها على الإختيارات التالية :

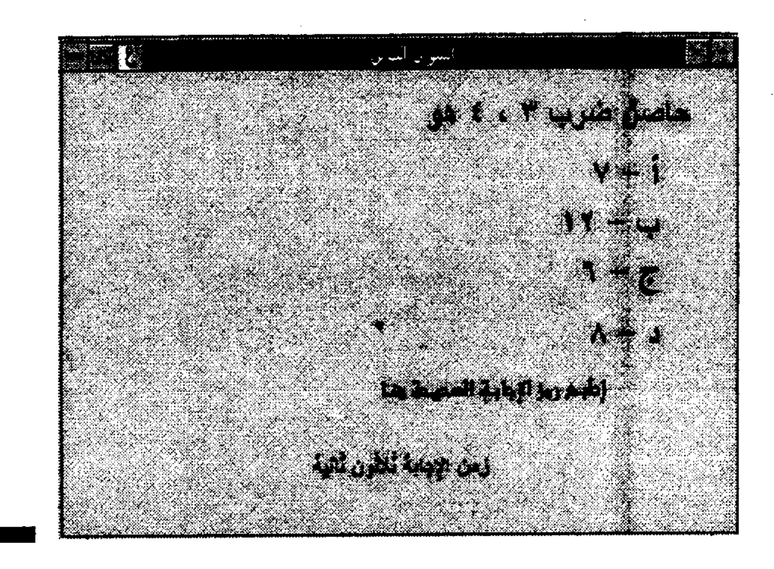
0 - None الخاصية لاتعمل
1 - Inset للداخل
2 - Raised للخارج

- ✍ خاصـــية *Outline* لتحديد حدود البنل من الخـــــارج من عدمه True / False
- ✍ خاصـــية *RoundedCorners* لتحديد استدارة حـــواف البـــنل من عـــدمه True/False

- ✍ خاصـــية *ShadowColor* لتحديد لون سمك البنل أى كان بارزا أو غائرا والتى تشتمل بدورها على الإختيارين التاليين :

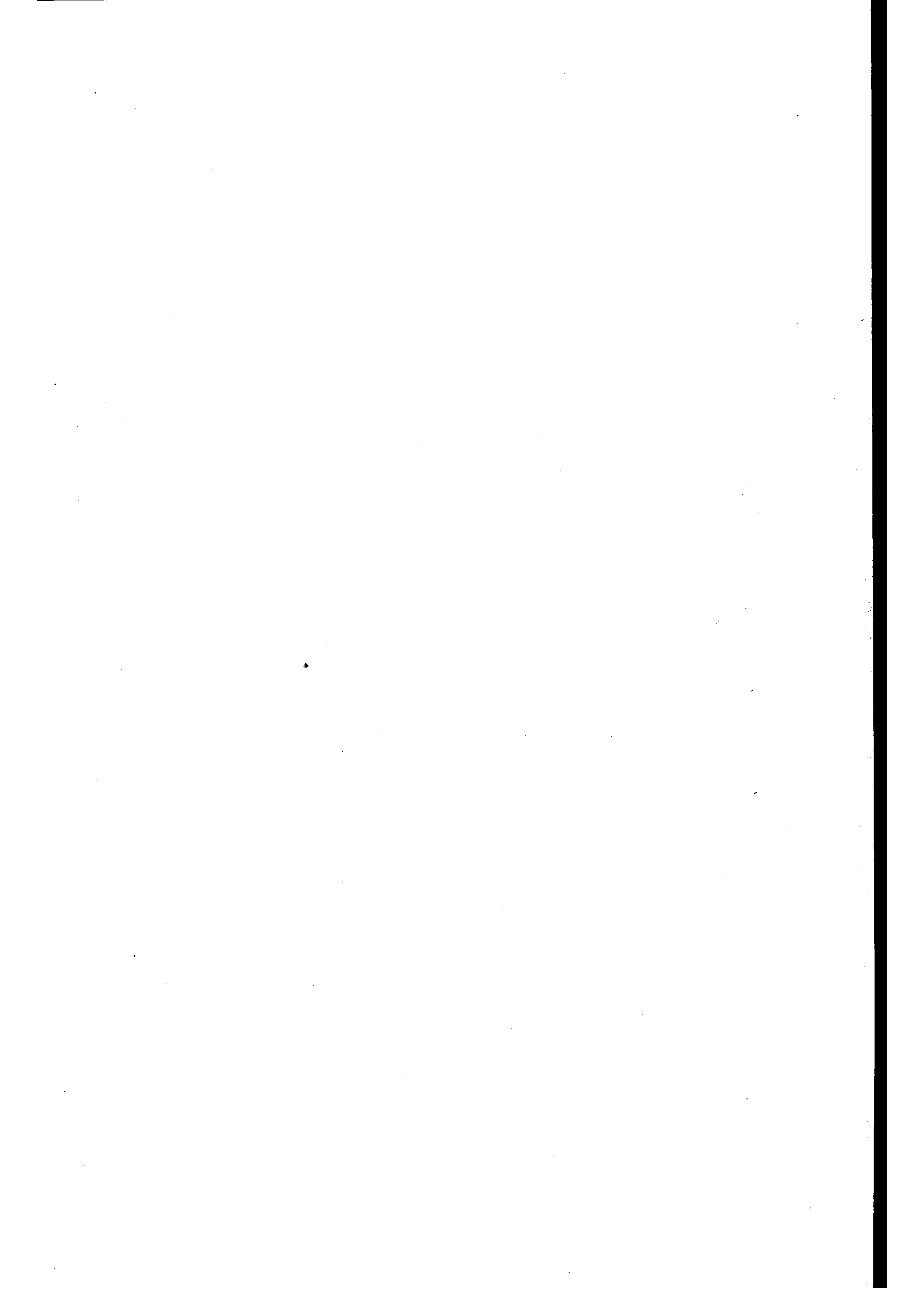
0 - Dark Grey رصاصي غامق
1 - Black أسود

- ✍ خاصـــية *BackColor* لتحديد لون خلفية البنل .
- ✍ خاصـــية *ForeColor* لتحديد لون حـــروف الرسالة المكـــتوبة على سطح البنل .



الفصل الثامن

تصميم وإنتاج شاشات التمارين والاختبارات

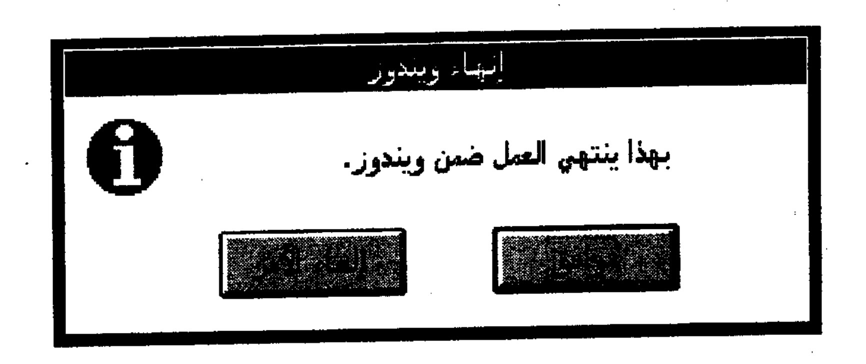


تمهيد

يعتبر تصميم التدريبات والتمارين والإختبارات من أهم المضامين التربوية فى تصميم البرمجية التعليمية ، وخاصة اذا ماكانت اختبارات تشخصية أو اختبارات تسكين (اختبارات تحديدي مستوى) . ان التدريبات والتمارين وبعض انواع الإختبارات لاتطلب حفظ درجة أداء المتعلم ومعالجتها لتقييمه ؛ بقدر ماتتطلب توفير تغذية راجعة فورية وجيدة تقدم للمتعلم عند اجابته الصحيحة والخاطئة على حد سواء. هذا وتهتبر صناديق الحوار التى تتميز بها بيئة بيسك المرئية من اهم الأدوات المعينة للمبرمج على تقديم تغذية راجعة ، جيدة وفورية للمتعلم ، والتى سوف نتناولها بمزيد من التفصيل فيما يلى :

صناديق الحوار Dialog Boxes

قد يتطلب الأمر فى كثير من التطبيقات ارسال رسالة عاجلة الى مستخدم البرنامج (المتعلم) ، ويختلف مضمون الرسالة بحسب الموقف ، فقد تكون رسالة تنبيه كالرسالة التى تظهر على الشاشة عندما تغلق نافذة مدير البرنامج (Program Manager) حيث تنبهك الرسالة الى انك فى طريقك الى مغادرة بيئة وندوز وتطلب منك تأكيد رغبتك فى الخروج بالضغط على زر الأوامر < **OK موافق** > أو على الزر < **Cancel إلغاء الأمر**> للعودة الى وندوز مرة اخرى .



كما تستخدم الرسائل ايضا لنقديم تغذية راجعة جيدة ومفيدة للكتعلم اثناء التدريب والمران ، وبالطبع – كما سبق وان استعرضنا – ان هناك بعض

الصناديق مثل صناديق النصوص والقوائم التى تستخدم فى استقبال البيانات ، ولكن صناديق الحوار بصفة عامة لها اهمية خاصة ، فهى عبارة عن **نوافذ قافزة تظهر على الشاشة فى موقف معين وتطلب من المستخدم ادخال بيان ما أو الإستجابة لحدث ما بحيث لايمكن أن يستمر البرنامج بدون الإستجابة لهذه النافذة.** وبالتالى فهى تعتبر مفيدة للغاية للمبرمج كأدوات سابقة التجهيز لتقديم التغذية الراجعة الفورية للمتعلم .ولدينا فى نظام بيسك المرئي ؛ نوعان من صناديق الحوار هما : صناديق الرسائل ***Massage Boxes*** وصناديق الدخل ***Input Boxes***

(أ) صناديق الرسائل Massage Boxes

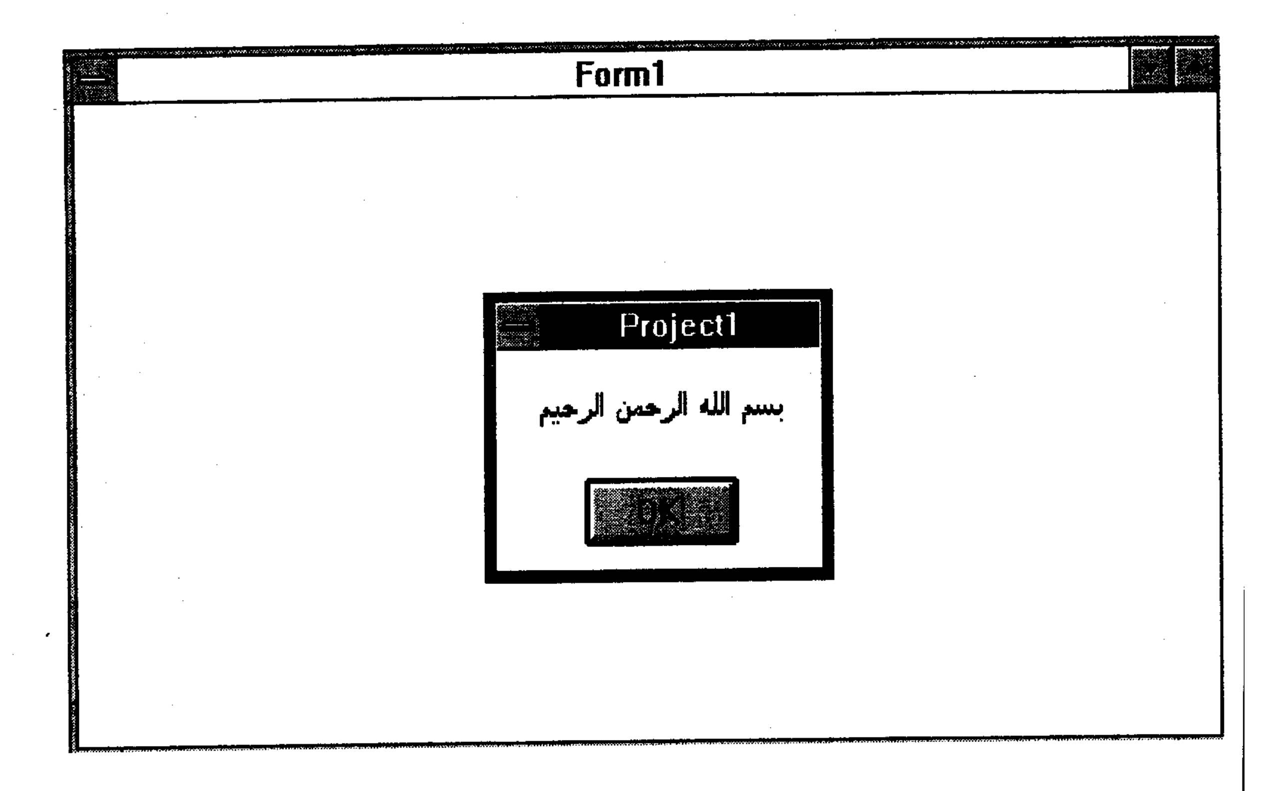
هى احدى انواع صناديق الحوار ، وبالتأكيد سوف تحتاج الى استخدام تلك الصناديق فى كثير من المواقع بمشروعك ، ولفهم ماهية صندوق الرسالة MsgBox نفذ التدريب التالى :

تدريب (٩)

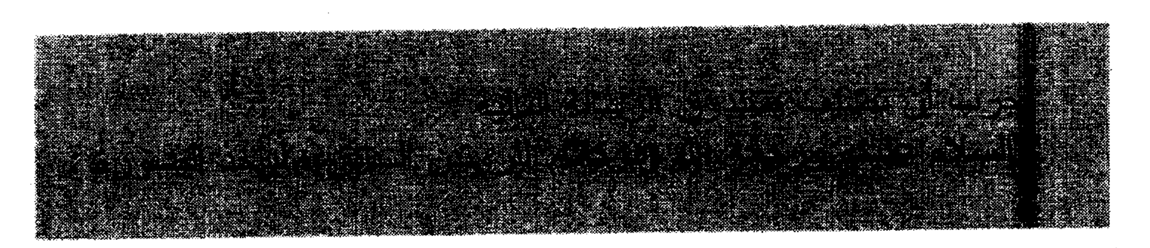
ابدأ مشروعا جديدا ، اكتب العبارة التالية فى روتين النقر على النموذج (Form_Click) : ليأخذ هذا الروتين الصورة التالية :

```
Sub Form _ Click ()
   MsgBox " بسم الله الرحمن الرحيم "
 End Sub
```

نفذ المشروع ، ثم انقر على النموذج بالفأرة سوف يظهر على الفور نافذة الرسالة والتى يحتوى على صندوق الرسالة المطلوبة " بسم الله الرحمن الرحيم " مع وجود الزر < OK < ، وصندوق قائمة التحكم بصورة سابقة التعريف Default وبدون اى جهد خاص من جهة المبرمج ، وسرعان مايختفى صندوق الرسالة بمجرد الضغط على زر < **OK** < ؛ انظر الشكل التالي :



هذا ويمكن اضافة أكثر من صندوق رسالة لنفس الروتين اذا ما استهدف المبرمج ذلك ، وفى هذه الحالة سوف يتم ظهور صناديق الرسائل بشكل متتالي على النموذج حسب اولوية وضع الصناديق داخل الروتين المستهدف .



```
Sub Form _ Click ()
   MsgBox  "بسم الله الرحمن الرحيم"
   MsgBox  "السلام عليكم ورحمة الله وبركاته"
End Sub
```

ملاحظة ١

ان الأمر MsgBox يتسبب فى طباعة التعبير الحرفى الذى يليها ، والذى ينبغى ان يوضع بين علامتى التنصيص " " ، وكما نرى ان النافذة تأخذ الأبعاد المناسبة للرسالة المكتوبة بداخلها ، ولو أنك كتبت عبارة اطول فسوف تستطيل النافذة بالقدر المناسب.

ملاحظة ٢

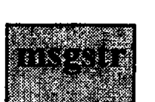
يحتوى صندوق قائمة التحكم على اختيار واحد فقط وهو الإختيار (نقل Move) لتحريك النافذة من مكانها ، وهذا كل مايلزم لنافذة الرسالة.

وعندما يظهر صندوق الرسالة على الشاشة فلابد من ان تضغط على الزر < **OK** > حتى يختفى الصندوق ، حيث لاتستطيع بحال ان تتخطاه وتستمر فى عمل آخر.

اضافات مفيدة لصندوق الرسائل

ان الصورة السابقة لعبارة صندوق الرسالة هى ابسط صورة ممكنة ولكنك مع ذلك تستطيع ان تضيف الى الصندوق المزيد من الملامح ، وهذه هى الصورة العامة لعبارة صندوق الرسالة :

```
MsgBox "msgstr", options , "titlestr"
```



التعبير الحرفى المطلوب عرضه بالصندوق ، وهذا هو الجزء الإجبارى فى العبارة أما عدا ذلك فهو اختيارى.


تعبير حرفى يستخدم كعنوان انافذة صندوق الرسالة ، بـــدلا مـــن التعبير المبدئي Project1 كما ظهر فى التدريب السابق.



تعبير عددى يحدد صفات النافذة المطلوبة مثل الأيقونات والأزرار



جرب الآن تعديل كود التدريب السابق ليصبح على الصورة :

```
Sub Form _ Click ()
   MsgBox  "الإجابة خاطئة : حاول مرة اخرى" , 49 , "السؤال الأول"
End Sub
```

سوف تشاهد صندوق الرسالة بالصورة الآتية :



لقد ظـــــهرت عبـــــارة "السـؤال الأول" كعنوان للصندوق ، كما تسبب العدد 49 فـــى ظهور الزرين معا بالنافذة علاوة على أيقونة علامة التعجب.

ان العدد 49 وغيره من الأعداد عبارة عن ثوابت سبق تعريفها في لغة بيسك المرئية وهى جميعا موجودة بالملف CONSTANT.TXT

وظائف الثوابت المتضمنة بملف الثوابت CONSTANT.TXT

تستخدم الثوابت لتحديد الخصائص التالية للصندوق :

١- انواع الأزرار المطلوبة بالصندوق التى تظهر في صورة مجموعات بإستثناء المفتاح السابق التعريف ؛ انظر الجدول رقم (١):

٢- تحديد الزر سابق التعريف Default : حيث يظهر الزر سابق التعريف محاطا بإطار ويستجيب للضغط على مفتاح Enter مثل زر < OK > فى التدريب السابق؛ انظر الجدول رقم (٢) :

٣- رسم ايقونة فى الصندوق ؛ انظر الجدول رقم (٣) :

٤- تحديد أولوية الصندوق ؛ انظر الجدول رقم (٤) :

والجداول (١) ، (٢) ، (٣) ، (٤) : توضح هذه الثوابت ، ونرى بالجداول ان لكل ثابت قيمة عددية واسما ، وهذا ويجوز استخدام أسماء الثوابت بدلا من القيم العددية بعد تحميل الملف CONSTANT.TXT ضمن المشروع العامل.

جدول (١)
الثوابت التى تحدد انواع ومجموعات الأزرار

[illegible]	[illegible]	[illegible]
MB - OK	0	لعرض الزر OK
MB - OKCancel	1	لعرض الزرين : Cancel OK
MB - AbortRetryIgnore	2	لعرض الأزرار الثلاثة: Ignore Retry Abort
MB - YesNoCancel	3	لعرض الأزرار الثلاثة: Cancel No Yes
MB - YesNo	4	لعرض الزرين : No Yes
MB - RetryCancel	5	لعرض الزرين : Cancel Retry

جدول (٢)
الثوابت التى تحدد ترتيب الزر سابق التعريف

[illegible]	[illegible]	[illegible]
MB - DEFBUTTON1	0	الأول
MB - DEFBUTTON2	1	الثانى
MB - DEFBUTTON3	2	الثالث

جدول (٣)
الثوابت التى تحددشكل الأيقونة

مسمى الثابت	قيمة الثابت	الأيقونة الناتجة
	0	لاشئ
MB – ICONSTOP	16	أيقونة علامة STOP قف
MB – ICONQUESTION	32	أيقونة علامة ? استفهام

(تابع) جدول (٣)
الثوابت التى تحددشكل الأيقونة

مسمى الثابت	قيمة الثابت	الأيقونة الناتجة
MB – ICONEXCLAMATION	48	أيقونة علامة تعجب
MB – ICONINFORMATION	64	أيقونة تحتوى على الحرف الأول من كلمة Information

جدول (٤)
الثوابت التى تحدد أولوية الصندوق

مسمى الثابت	قيمة الثابت	الوظيفة
MB – APPLMODAL	0	يحتل الصندوق أولوية بالنسبة للتطبيق الحالى فقط بمعنى انه يمكنك تشغيل مدير الملفات مثلا قبل الإستجابة للصندوق.

(تابع) جدول (٤)
الثوابت التى تحدد أولوية الصندوق

[illegible]	[illegible]	[illegible]
MB – SYSTEMMODAL	4096	يحتل الصندوق أولوية مطلقة على جميع التطبيقات فلا يمكنك الإنتقال الى اى تطبيق قبل الإستجابة الى صندوق الرسالة.

وقد رأينا من خلال تدريب (١٠) ان الروتين الفرعى التالي : قد انتج صندوقا يحتوى على ايقونة علامة التعجب وعلى الزرين **< OK >** ، **< Cancel >**

```
Sub Form _ Click ()
   MsgBox  "السؤال الأول" , 49 , " الإجابة خاطئة : حاول مرة اخرى"
End Sub
```

حيث ان العدد (49) يعبر عن مجموع الثوابت الآتية :

48	ثابت أيقونة علامة التعجب	MB - ICONEXCLAMATION
1	ثابت مجموة الأزرار	MB - OKCANCEL
0	ثابت الزر سابق التعريف	MB - DEFBUTTON1

هذا ويلاحظ انه عند وضع القيمة (0) للثابت الثالث ؛ يصبح الزر الأيسر هو الزر سابق التعريف Default

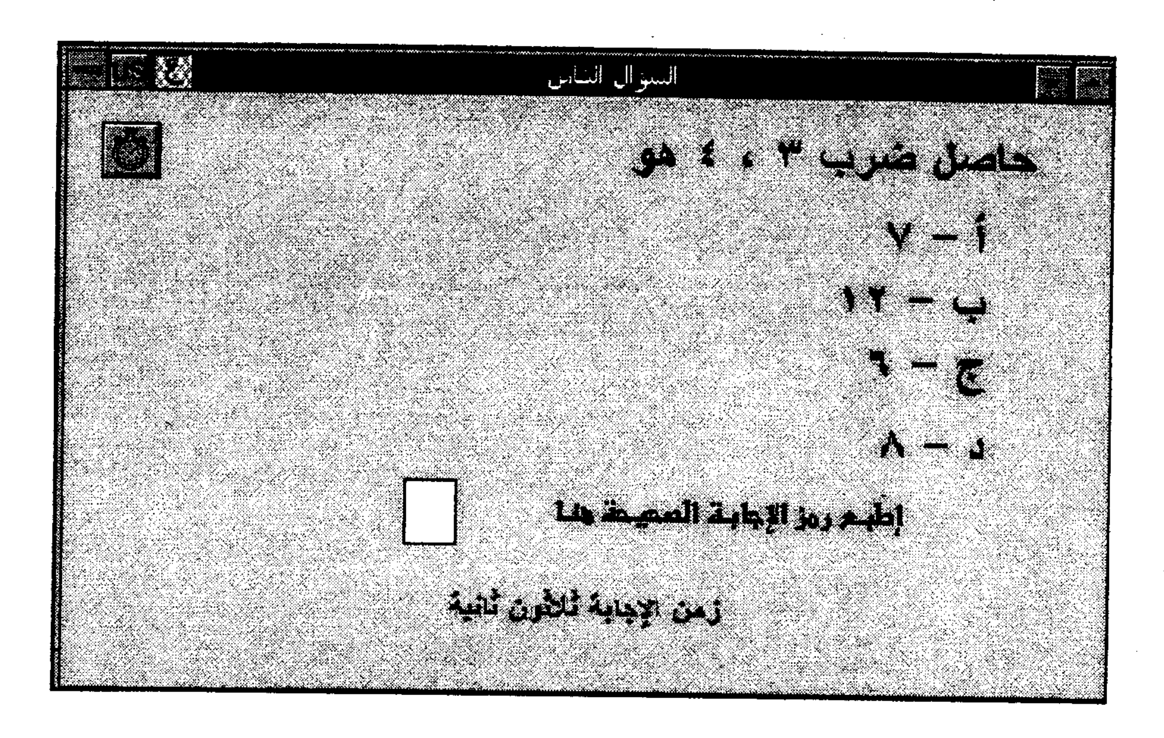
هذا ويمكننا الإستفادة من صناديق الرسائل لتقديم تغذية راجعة جيدة حالة اجابة المتعلم على التدريبات والتمارين اجابات صـــحيحة أو خاطئة مثل :

ممتاز : اجابتك صحيحة
الإجابة صحيحة : احسنت عملا
بارك الله فيك : فإجابتك صحيحة
أجبت صح : أكثر الله من امثالك

الإجابة خاطئة : حاول مرة أخرى
لا : هذه ليست الإجابة الصحيحة
أجبت خطأ : فكر معى مرة اخرى
الإجابة خاطئة : ليست ٣+٧=١٠

تدريب (١٨)

يتكون التدريب التالي من نافذة واحدة تحتوى على مجموعة من وحدات التحكم لإضافة النصوص كلاواصق **Label** ؛ ووحدة تحكم أخرى لإضافة النصوص كنصوص **Text** ، بالإضافة إلى ساعة للتحكم في زمن العرض **Timer** (الزمن المخصص للسؤال) ، كما هو واضح من الشكل رقم (٢٢):



شكل رقم (٢٢)

شاشة مفردة اختبار من نوع
اختيار من متعدد



وذلك بفتح النافذة المنسدلة للملفات **File** واختر مشروع جديد ، سوف يظهر على الفور شاشة (واجهة) المستخدم User's Interface بالمسمى المبدئي Form1

حدد خصائص النموذج Form1 باستخدام نافذة الخصائص Properties Windows والفأرة كما سبق لتصبح كماهية موضحة بالجدول التالي :

خصائص Form1		
Control	**Property**	**Value**
Form1		
	Captions	السؤال الثامن
	BackColor	&H00FF00FF& اللون الأصفر
	Height	5550
	Left	1440
	Top	1100
	Width	7500

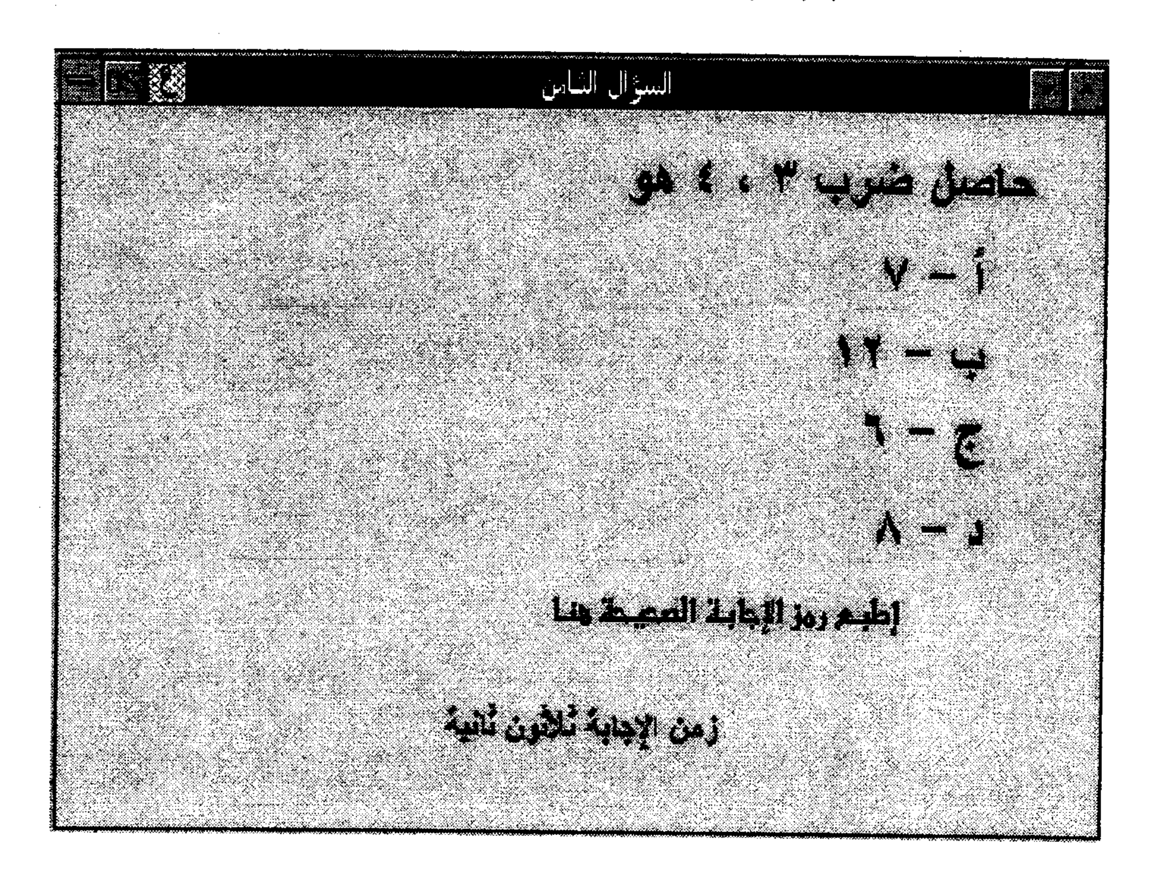
لاحظ عمل الخواص التالية :

للنموذج	لصندوق النصوص
ForeColor	
لتحديد لون نسيج (نقاط) الخلفية	لتحديد لون الحروف بالصندوق
BackColor	
لتحديد لون خلفية النموذج	لتحديد لون خلفية الصندوق

التعامل مع صناديق اللواصق Label Box [illegible]



أضف صناديق اللواصق السبع المطلوبة بواجهـــة المســتخدم ، وذلك بالنقر على أداة اللواصق ؛ حدد خصائصها باســتخدام الفــأرة ونافذة الخصائص كما سبق ؛ لتصبح واجهة المســـتخدم كمـــا هـــي موضحة بالشكل رقم (٢٣) :



شكل رقم (٢٣)

شاشة مفردة اختبار موضحا عليها النصوص كلاواصق

[illegible] Text Box [illegible]

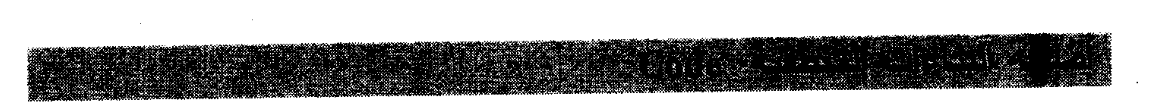
تعمل أداة إضافة صندوق النصوص **Text Box** على عـرض صندوقـا لنص يمكن المتعلم من إدخال نص به ، هذا ويجب أن يكون حجم الصنـدوق مناسبا لحجم النص المتوقع إدخاله بواسطة المتعلم.

أضف صندوق النصوص المطلوب بواجهـة المسـتخدم ، وذلك بالنقر على أداة النصوص ؛ سوف يظهر لـك صنـدوق النصوص على شاشة النموذج (واجهة المستخدم) بالاسم المبدئي **Text1** ، اسحب الصندوق بالفأرة إلى الموقع المناسـب بـالنموذج ، لجعـل الصندوق فرغا ليستطيع المتعلم إدخال النص المطلوب : غير قيمة الخاصيـة *Text* من *Text1* إلى فراغ.

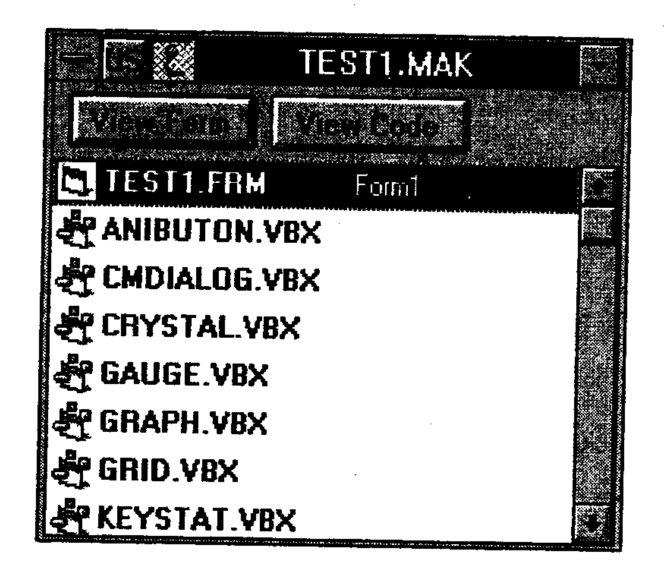
التعامل مع ساعة التحكم Timer

تعمل أداة ضبط الوقت على التحكم في زمن عرض النوافـذ أو لتصيـد الأحداث *Event Trapping* في فترة زمنية محددة ، وتعرض أثنـاء مرحلـة التصميم ولا يعرض أثناء مرحلة التشغيل .

لإضافة أداة ضبط الوقت إلى نافذة المشـروع : أنقـر أداة ضـبط الوقت سوف يظهر لك صندوق سـاعة التحكـم علـى النموذج ، اسحب صندوق الساعة بالفأرة إلى أقصى اليسار أعلا النموذج.

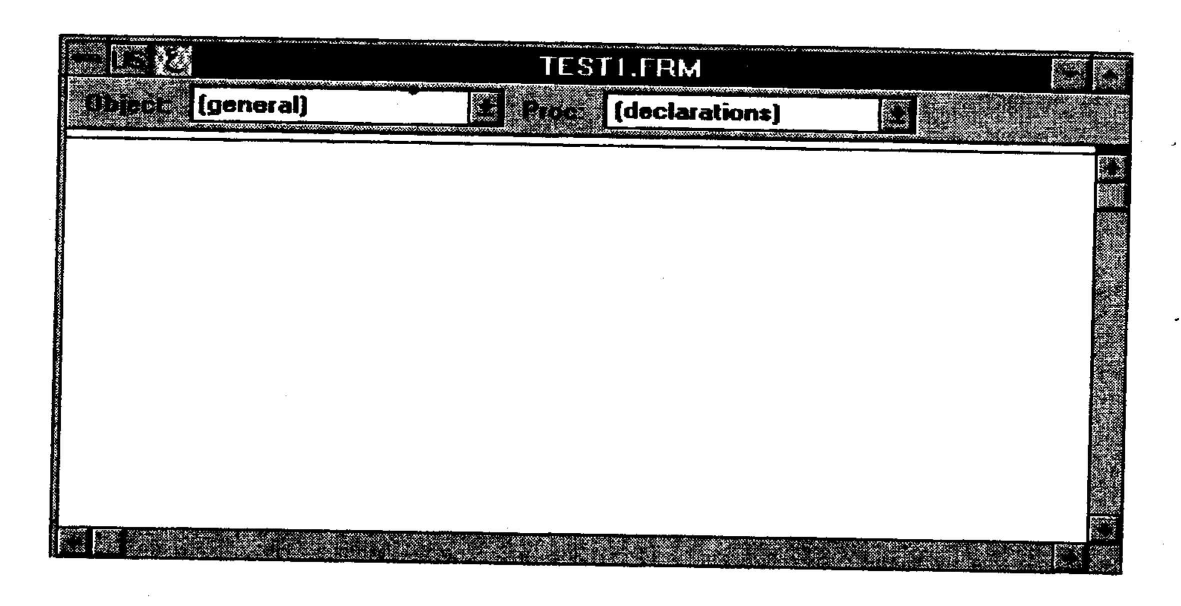


والآن علينا أن نضيف البنـاءات المنطقيـة اللازمـة للمشروع ، فمن نافذة المشروع الموضحة بالشكل رقم (٢٤) تخـير سـوف يقودك النظام إلى شاشة الكود والتي تتكون بدورها من عدد من الصنـاديق والنوافذ ، والموضحة بالشكل رقم (٢٥)



شكل رقم (٢٤)

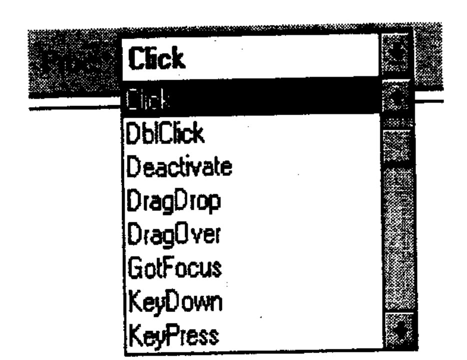
شاشة المشروع



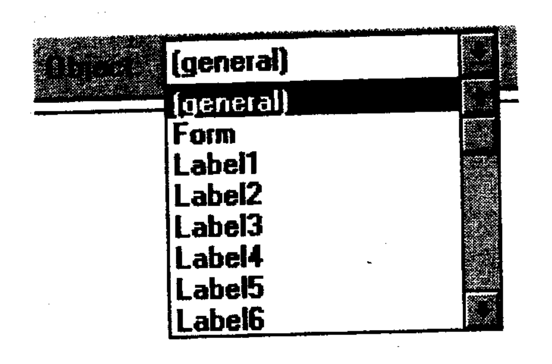
شكل رقم (٢٥)

شاشة الكود

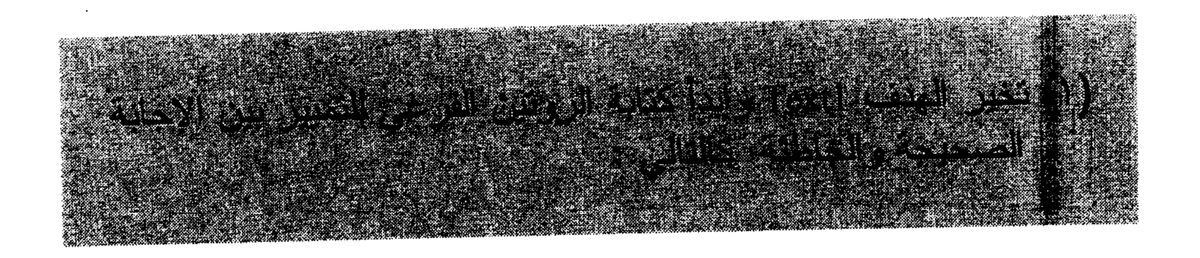
تخير منها الهدف المناسب وذلك بفتح نافذة الأهداف Object ، والتي عادة ما تحتوى على كافة الأهداف (الأشياء) المدرجة بمشروعك مثـــل : , *Form* *Label1 , label2 , Label3 , Label4 , Label5 , Label6 , Label7, Text1, Timer1* وبالإضافة إلي (*general*). وكذلك تخير الإجـــراء المناســـب بفتـــح نـــافذة الإجراءات والتي عادة ما تحوى على كافة الإجراءات التي قد تحتاجها للهدف الذي تم اختياره مثل : (*Activate , Click , DblClick , ...*)



نافذة الإجراءات



نافذة الأشياء



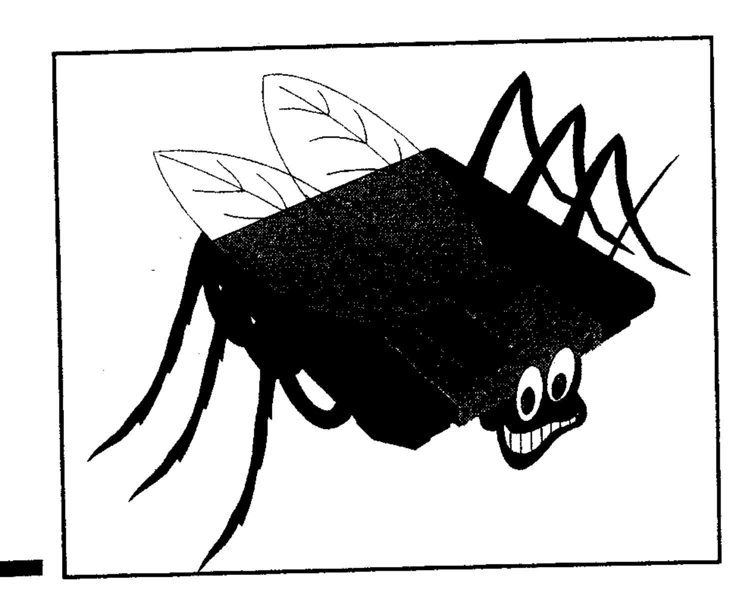
( ) تخير الهدف (Text1) وابدأ كتابة الروتين الفرعي للتمييز بين الإجابة الصحيحة والخاطئة كالتالي :

```
Sub Text1_Change ()
   Dim answer
   answer = "ب"
   If text1.Text = answer Then
   MsgBox "الإجابة صح": MsgBox "انتهى السؤال"
   Else MsgBox "الإجابة خطأ"
   MsgBox "انتهى السؤال"
   End If
End Sub
```

(٢) تخير الهدف Timer1 وابدأ كتابة الروتين الفرعي لضبط الساعة لمدة ثلاثين ثانية هي زمن الإجابة على السؤال كالتالي :

```
Sub Timer1_Change ()
   Dim dflvalue
   dflvalue = 10000
   If timer1.Interval = dftvalue Then
   Form1.Show 1
   End  If
End Sub
```

لاحظ أنك قد لاحتاج إلي تعديل بقية الخواص حيث تترك كما هي بقيمها المبدئية Default ، وعليك تعديل الخواص التي تؤدى إلى تحسين عملك مرحليا فقط .



الفصل التاسع

التعامل مع الصور وتحريكها

تمهيد

عرفنا فيما سبق كيفية إضافة صورة سابقة الإعداد إلى واجهة المستخدم ، حيث أن نظام البيسك المرئي يمكننا من إضافة ثلاثة أنواع من الصور ، حيث يتم التمييز بينهما بامتداد اسم الملف .BMP, .ICO, .WMF

هذا ويجوز تضمين الصور عن طريق أدوات إضافة الصور:


صندوق الصور Picture Box أو


صندوق الصور المبسطة (Image Box)

حيث تتميز صناديق الصور عن صناديق الصور المبسطة في أنها تستوعب الصور علاوة على أساليب الرسم المختلفة وخصائص Clip Picture

وقد عرفنا خلال المثال رقم (٢) أنه يمكننا إضافة صورة ما في صندوق الصور في مرحلة التصميم ، وذلك عن طريق الخاصية Picture من نافذة الخصائص (Custom. DIC).



والآن لإضافة الحركة إلى بعض الصور بالمشروع يفضل استخدام دالة تحميل الصور أثناء تشغيل البرنامج :

```
Load Picture ( Path  - name )
```

تدريب (١٢)

```
Picture 1 . Picture = load picture ( "c : \ windows \ leaves. BMP")
```

تؤدي هذه التعليمة إلى تحميل ملف الصورة " Leaves - BMP " صورة ورقة الشجر – والموجودة في برنامج ويندوز – في صندوق الصورة Picture 1

جرب استخدام هذه التعليمة بوضعها في روتين تحميل النموذج لمشروع جديد (بعد وضع صندوق صورة على النموذج).

```
Sub Form - Load ( )
  Picture1. Picture = Load Picture ("c:\windows\Leaves.BMP")
End Sub
```

يمكن بنفس الطريقة تحميل صورة في صندوق الصور المبسطة بالأمر التالي :

```
Image1. Picture = Load Picture ("c:\windows\Leaves .BMP")
```

خاصية تغيير المساحة آليا Auto Size

عند تحميل إحدى الصور في صندوق صور فان الصندوق لا يتغير مساحته مهما كانت مساحة الصورة ، فإذا كانت الصورة أكبر من الصندوق ظهر منها الذي يملأ الصندوق فقط : الصندوق (أ) ، أما إذا كانت الصورة أقل من الصندوق فتظهر الصورة كاملة بالصندوق مع ظهور فراغ بالصندوق

(ب) ، أما عند تعديل مساحة الصندوق تلقائيا لكي يتناسب مع مساحة الصورة المستقبلة عن طريق تحديد الخاصية Auto Size بالقيمة True فان مساحة الصندوق تتغير تلقائيا طبقا لمساحة الصورة المستقبلة : الصندوق (ج).

خاصية التغير الآلي للمساحة Auto Size مضبوطة على الوضع False بصفة مبدئية Default ، وهي من خصائص صناديق الصور فقط. أما صناديق الصور المبسطة فلا تحتاج أصلا إلى هذه الخاصية لأنها عاملة بصفة مستديمة.

جرب ذلك بتحميل نفس الصورة في صندوق الصورة Picture1 ، وفي صندوق آخر للصورة المبسطة Image1 ، وشاهد الفرق.

خاصية الانكماش والتمدد Stretch

توجد هذه الخاصية في صناديق الصور المبسطة فقط حيث تأخذ القيمة False بصفة مبدئية ، وبتعديل القيمة إلى True ، فان عملية تعديل المسافة الآلية سوف تعمل بطريقة مختلفة تماما ، فبدلا من أن يضبط الصندوق مساحته على قدر مساحة الصورة ، يحدث العكس ، فتتمدد الصورة ؛ تتكمش لكي تضبط نفسها في الصندوق. وبفضل هذه الخاصية يمكنك التحكم في مظهر الصورة تمددا أو انكماشا حسب الطلب.

تحريك الصور بأمر Move

بتعديل الخصائص Top ، Left للصندوق يمكنك من تحريك الصورة من موضعها فمثلا عن طريق التعليمة :

```
My Picture. Left = My Picture.left + 500
```

نزيد قيمة الإحداثي الأفقي Left بمقدار500 تويب (*)

وعن طريق التعليمة :

```
My Picture. Left = My Picture.Top + 200
```

نزيد من قيمة الإحداثي الرأسي (Top) بمقدار250 تويب (*)

أي أن الصورة المعنية بالحركة سوف تتحرك إلى اليمين وإلى أسفل بـــالمقدار المحدد سابقا

هذا واختصارا لسطور البرنامج ولسهولة التراكيب يمكن استخدام تعليمة Move كالتالي :

My Picture. ***Move*** Picture1 . ***Left***+500, Picture1. ***Top***+250

والصيغة العامة للتعليمة ***Move*** تأخذ الصورة :

---

* التويب Twip هو وحدة قياس الأبعاد ، وأن التويب الواحد يعادل ١/١٤٤٠ من البوصة.

Object.Move Left, Top, Width, Height

حيث

Object اسم النموذج أو الأداة المطلوب تحريكها (الهدف)
Left الإحداثي الأفقي للهدف
Top الإحداثي الرأسي للهدف
Width اتساع الهدف
Height ارتفاع الهدف

هذا وتعتبر جميع المتغيرات السابقة اختيارية فيما عدا المتغير Left

فمثلا التعليمة :

My Picture. Move Picture1. Left, Picture1, Top ,
2* Picture1. Width, 2*Picture1. Height

تعمل علي تحريك الصورة مع تغيير حجمها

تدريب (١٢)

يتكون هذا المثال من نافذة واحدة تحتوى على صندوق صورة مبسطة بالقرب من الركن الأيسر العلوي للنموذج ، وعند بدء التنفيذ يتم تحميل الصورة المحددة ، وتتحرك الصورة مع كل ضغطة بالفأرة إلى اليمين وإلى أسفل النموذج.

- ابدأ بمشروع جديد
- أضف صندوق الصور المبسطة Image Box .
- اسحبه إلى الركن الأيسر العلوي للنموذج
- من نافذة المشروع تخيير View Code ، وأكتب ما يلي:

روتين الضغط على النموذج

```
Sub Form - Click ( )
      Image1. Top = Image1. Top + 100
      Image1. Left = Image1. Left + 400
End Sub
```

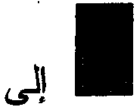
روتين تحميل النموذج

```
Sub Form - Load ( )
  Image 1. Picture + Load Picture ( "c:\windows\arches.bmp")
End Sub
```

عند تنفيذ المشروع سوف تظهر واجهة المستخدم وعليها صورة المنزل الصغير arches-bmp وسوف يتحرك المنزل إلى اليمين وإلى أسفل النموذج بقدر محدد مع كل ضغطة بالفأرة على النموذج .

لاحظ أنه عند تحميل الصور (صورة أو صورة مبسطة) يلزمنا تحديد اسم الملف الحاوي على الصورة ، وكذلك مسلك هذا الملف مثل :

("c:\windows\arches.bmp")



يفضل في هذه المرحلة للسهولة نقل الصورة المطلوب التعامل معها إلى القرص المرن الحاوي على المشروع قيد العمل.

تدريب (١٤)

يتكون المثال التالي من نافذة واحدة تحتوي على صندوقين للصور المبسطة : يحتوي الصندوق الأول على الصورة arches.bmp ؛ صورة المنزل والتي سبق لنا استخدامها بالمثال السابق ، ويحتوي الثاني على الصورة misc26.Ico ؛ وهي عبارة عن صورة سيدة ، لتوضيح شكل آخر لحركة الصور.

- ابدأ مع مشروع جديد New Project
- حدد خصائص النموذج Form1
- أضف صناديق الصور المبسطة ، اسحبها إلى مواقعها المناسبة على النموذج بالفارة.
- من نافذة المشروع تخير View Code.
- أكتب الروتينيات التالية :

١- روتين الضغط على النموذج :

```
Sub form - Click (   )
        Dim delta As Integer
         delta = 4
        Do
               image2.Move image2.Left - delta
        Loop Until image2.Left < 1320
End Sub
```

٢- روتين تحميل النموذج :

```
Sub Form- load ( )
```

ملاحظة ١

المتغير Delta متغير اختياري وقد أخذ القيمة 4 ، حيث أن السيدة تتحرك بمقدار Delta جهة اليسار حتى تبلغ المنزل.

ملاحظة ٢

استخدامنا -Do- Loop وهى أنسب أنواع الحلقات التكرارية لهذا الغرض.

ملاحظة ٣

تعتمد سلامة الحركة في هذا المشروع على قيمة المتغير Delta الذي يمثل خطوة الحركة فإذا كانت قيمة الخطوة أكبر مما يجب فإنها قد تؤدي إلى حركة سريعة مضطربة ولنحاول تغيير قيمة هذا المتغير حتى نحصل على أفضل أداء.

عند جلب صورة بحجم الشاشة يراعى ما يلي :

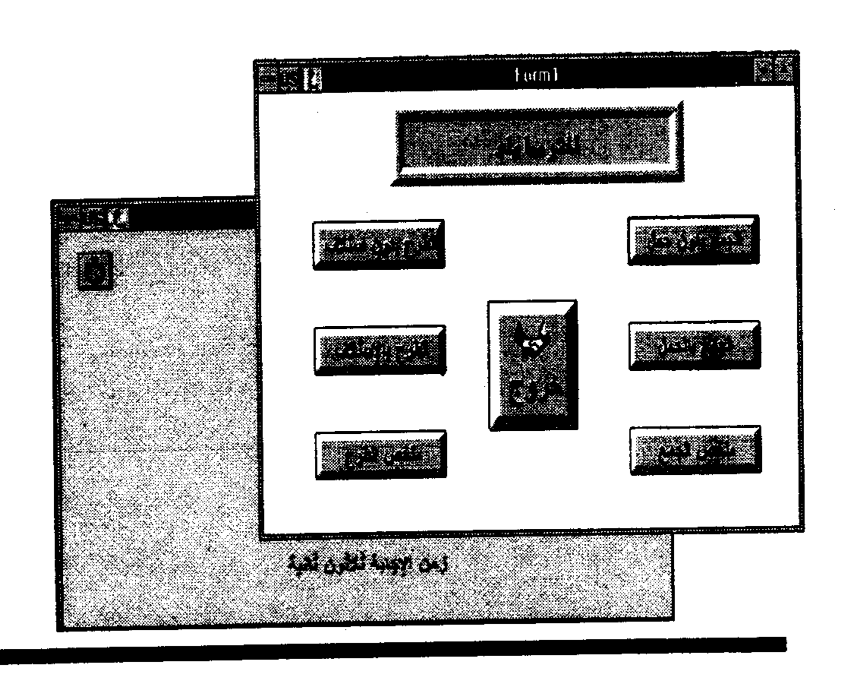
Shape 1 : Shape		
Black color		تحديد لون الإطار
Border Style	1- Sold	يحدد الإطار بخط متصل
Border Width	3	يحدد سماكة خط الإطار
Draw Mode	13- CopyPen	يحدد وضوح خط الإطار
Fill Style	1- TranSparent	يحدد ظهور الصورة مع الإطار
Fram 1		
Scale Left	0	

في آخر خطوة في المشروع
إجعل الشاشة بحجم شاشة الحاسوب وذلك عن طريق الخاصية
windowState = 2 - Maximized

آخر شاشة في الاختبار (Itel)

عند الضغط على الاستمرار تأتي الشاشة (Form 18) ، والتي فيها أكـــثر من نص ، وهنا ينبغي أن تتضمن ساعة توقيت ؛ على أن يكون كـــود هـــذه الساعة كالتالي :

```
Sub Timer1. Timer( )
   DIM dfivalue
      dfivalue = 10000
      If timer1. Interval = dfivalue  Then
   From 18. Hide
   From 19. Show 1
   End If
End Sub
```

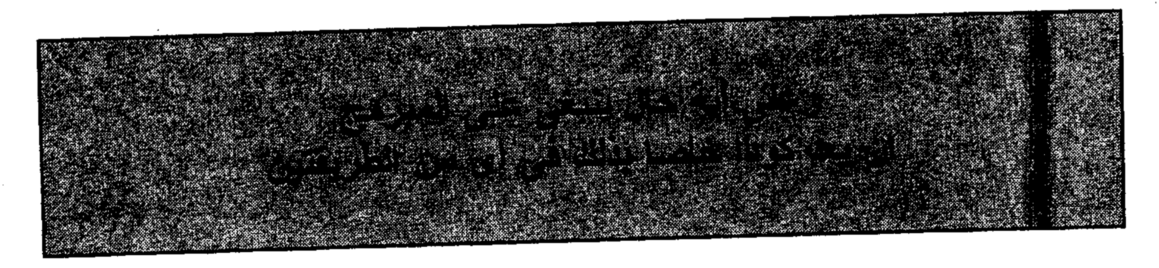


الفصل العاشر

كيفية الانتقال
من شاشة إلى أخرى

تمهيد

عادة ما يتم **الإنتقال من شـــــاشة الى أخرى** بإحدى الطريقتيـــن :
(١) **طريقة العداد** وهى الطريقة التى يتم عن طريقها الإنتقال من شاشة الـــى اخرى بعد إنقضاء فترة زمنية محددة وهو مايعرف بـــ ***Time action*** حيـث لايملك المتعلم التحكم فى هذا الإجراء وعادة ماتستخدم هذه الطريقة للإنتقــال من شاشة الى اخرى من الشاشات الغير هامة لتعلم المتعلم كشاشات المقدمـــة وشاشات التعريف بالمنتج . (٢) **الطريقة التى يتم عن طريقها الإنتقال مـــن شاشة الى اخرى عن طريق المتعلم** وبالتالى يكون المتعلـــم هـــو المتحكـم الأساسى والوحيد فى الإنتقال من شاشة الى اخرى أى من فقرة من فقـــرات الدرس الى فقرة اخرى أو من درس الى آخر أو من اختيار الى اختيار آخـــر فمثلا هناك من المتعلمين من يفضل الإنتقال من الشرح الـــى الإختبـــار دون المرور على الملخص أو حل التمارين ، ومنهم من يفضل ان يبدأ بـــالملخص وينتهى بالإختبار ... وهكذا ، وهنا ينبغى على المتعلم ان يضغط علـــى زر خاص قد اعده له المبرمج او ان يضغط على رقم او حرف على الشاشـــة ، وهذا يكفى بان ينتقله الحاسوب الى الجزء المطلوب فورا وبصورة مباشـــرة ، وهذا الإجراء هو مايعرف بـــ ***Click action***



بفرض انه لدينا فى البرمجية الحالية عدد من شاشات المقدمة (Form1 , Form2 , Form3) ، وشاشة المنتج (Form4) ، وشاشة اخرى للأهداف العامة للبرمجية (Form5)، وشاشة القائمة (Form6) ، وهكذا بقية شاشات البرمجيـــة

من الأفضل بالطبع ان يتم الإنتقال من Form1 الى Form2 ، ومن Form2 الى Form3 ، ومن Form3 الى Form4 ، ومن Form4 الى Form5 : باستخدام ساعة تحكم Timer ، أى سوف يتم الإنتقال بطريقة العداد *Time action* ، حيث ان تلك الشاشات الأربع ليس لها علاقة بتعلم المتعلم ، وبالتالى فهو ليس فى حاجة الى ان يتحكم بنفسه فى الأنتقال من شاشة الى اخرى . بينما يحتاج المتعلم الى الوقوف قليلا عند الشاشة الخامسة ليقرأ منها ويستوعب الأهداف العامة للبرمجية ، ويقررمتى ينتقل الى الشاشة التالية (Form6) ؛ وبالتالى فانه من الأفضل ان يتحكم المتعلم بنفسه فى توقيت الانتقال من الشاشة الخامسة (Form5) الى الشاشة السادسة (Form6) بحدث من صنعه هو *Click action* ..... وهكذا تتم عمليات الإنتقال من شاشة الى اخرى لبقية شاشات البرمجية .

وهنا سوف نحتاج الى وضع ساعة تحكم (عداد Timer) الى النموذج الذى سوف ننتقل منه الى النموذج التالى بطريقة العداد *Time action* بينما نحتاج الى وضع زر اوامر عادى Command button او زر اوامر مجسم 3D Command button الى النموذج الذى سوف ننتقل منه الى النموذج التالى بحدث من المتعلم *Click action* قبل البدء فى كتابة الكود اللازم لذلك.

طرق الإنتقال من شاشة إلي أخرى

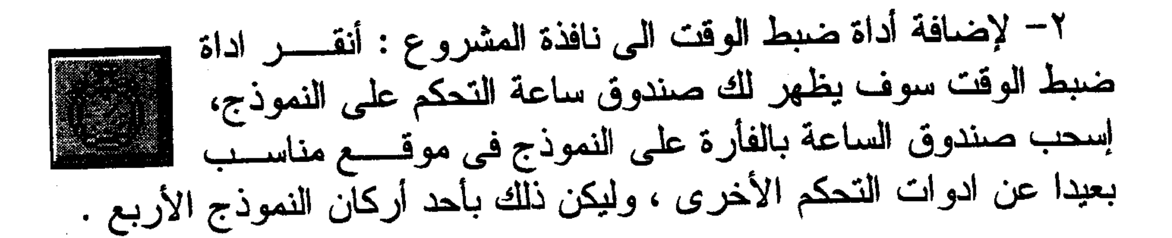
أولا : طريقة العداد ***Time Action***

يستخدم العداد (اداة ضبط الوقت) لحساب الوقت اثناء تنفيذ البرنامج ، ويمكننا ضبطه بحيث انه بعد مرور فترة زمنية بعينها يقوم بتوليد حدث معين لتشغيل انشطة اخرى ، انه يحاكى " المنبه " الذى يضرب الجرس فى ساعة معينة نضبطه عليها، وقد يتصل المنبه بجهاز راديو مثلا ، فيؤدى حدث الجرس الى تشغيل جهاز الراديو. زمنية محددة ، وتعرض أثناء مرحلة التصميم ولا يعرض أثناء مرحلة التشغيل .



التعامل مع العداد Timer لتحديد وقت ظهور الشاشة

١- تخير الشاشة المستهدفة ولتكن الشاشة الأولى من شاشات المقدمة (Form1)



٢- لإضافة أداة ضبط الوقت الى نافذة المشروع : أنقر اداة ضبط الوقت سوف يظهر لك صندوق ساعة التحكم على النموذج، إسحب صندوق الساعة بالفأرة على النموذج فى موقع مناسب بعيدا عن ادوات التحكم الأخرى ، وليكن ذلك بأحد أركان النموذج الأربع .

٣- حدد خصائص ساعة التحكم من خلال صندوق الخصائص الخاص بتلك الوحدة والموضح بالشكل رقم () ، والذى عادة ماستجدة محدودا للغاية فى خواصه . وحيث ان اداة العداد تتميز بمساحة ثابتة على الشاشة ، لذلك فان خصائص الطول Height والعرض Width لاينطبقا عليها وبالتالى فهما غير موجودان بالجدول.

Properties

Timer1 Timer

0

Enabled	True
Index	
Interval	0
Left	3120
Name	Timer1
Tag	
Top	1800

نافذة خواص العداد

خاصية Interval

تحدد قيمة هذه الخاصية الفترة الزمنية التى يجب ان تنقضى قبل ان يبدأ العداد فى توليد الحدث المطلوب منه ، وتقاس الفتره الزمنية بالميللى ثانية (الميللى ثانية تساوى جزء من ١٠٠٠ جزء من الثانية) وهى تبدأ من صفر وتصل الى ٦٥٥٣٥ ؛ والقيمة صفر تعنى تبطيل العداد اما القيمة العظمى فهى تتعدى الدقيقة بست ثوان تقريبا ، ومعنى هذا انه لايجوز ضبط العداد على فترات زمنية طويلة ، ويفضل ان تكون مدة ضبط العداد فى حدود ٣٠٠٠ اى ثلاثين ثانية ، وهى مدة كافية لروئية شاشة المقدمة :

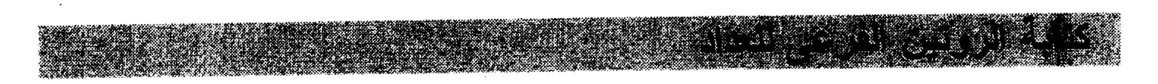
```
Timer1.Interval=30000
```

خاصية Enabled

تأخذ هذه الخاصية القيم المنطقية True / False،

True	تشغيل	False	ايقاف

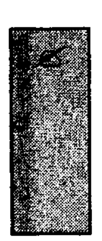
والقيمة سابقة التعريف Default بجدول الخصائص هى القيمة True والتى تجعل العداد يؤدي وظيفته أما القيمة False فتؤدى الى تبطيل العداد.



لكتابة الروتين الفرعى للعداد اتبع الخطوات التالية :



من نافذة المشـــــــروع أختر النـــافذة المســتهدفة (Form1) ، واضغط على زر (Veiw Code)



من نافذة الكود اختر من صندوق الأهداف (الهدف Timer1) ، سـوف تجد ان (الحدث Timer) قد ظهر تلقائيا فى صندوق الأحداث ، وعلـى الفور سيظهر لك بالجزء الخاص بتحرير الكـــود طرفـى الروتيـن الفرعى بالصورة:

```
Sub Timer1_Timer ()
End Sub
```

اكتب جسم الروتين الفرعى المطلوب ليأخذ الصورة التالية :

```
Sub Timer1_Timer ()
   If timer1.Interval = 30000  Then
      Form2.Show 1
   End  If
End Sub
```

وهو بالطبع يكافئ الروتين الفرعى التالى :

```
Sub Timer1_Timer ()
      Dim dflvalue
      dflvalue = 30000
       If timer1.Interval = dftvalue Then
          Form2.Show 1
       End  If
End Sub
```



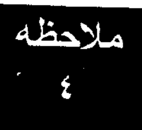
لاحظ ان ايعاز Dim اختيارى فى هذه الحالة لبساطة العمل وبالتالى يمكن عدم ذكر هذا الأمر.



لاحظ ان ايعاز dftvalue=30000 اسم متغير اختيارى لتسكين الفترة الزمنية للعداد ، ويجب هنا ان تكون القيمة المحددة للفترة الزمنية هى نفسها التى تم تحديدها فى خاصية Interval بصندوق خواص العداد.



لاحظ ان امر (ايعاز) Show 1 هو الأمر الذى سوف يتسبب فى اظهار النافذة Form 2 المستهدف ظهورها بعد انقضاء الفترة الزمنية المحددة.



لاحظ ان النافذة المستهدف ظهورها (Form2) بعد انقضاء الفترة الزمنية التى سبق تحديدها سوف يتم ظهورها فوق الشاشة الأولى (Form1).

بعض الضوابط الحاكمة لظهور الشاشات المستهدفة

باستخدام التكنيك السابق فى اظهار الشاشة المستهدف ظهورها ، تظهر الشاشة (Form2) فوق الشاشة (Form1) أى ان الشاشتين تبقيان عاملتان فى ذاكرة الحاسوب المؤقته RAM ، واذا استمر هذا مع بقية شاشات المشروع فإننا سوف نواجه حتما مشكلة ذاكرة غير كافية not enought memory ، هذا وعند الخروج من الشاشة (Form2) لأى سبب من الأسباب سوف تظهر الشاشة الأولى (Form1) ، وقد يكون هذا غير مستهدف من المبرمج. **ولحل هذه المشكلة يمكننا استخدام اى من الأمرين التاليين :**

Unload Form1	أو	Form1.Hide

إيعاز Form1.Hide

ويتسبب هذا الإيعاز في اخفاء الشاشة الأولى (العاملة Form1) من شاشة الحاسوب، مع بقائها بذاكرة الحاسوب المؤقته ، وبالتالى تبقى مشكلة التعرض لذاكرة ممتلئة قائمة، اضافة الى صعوبة استدعاء هذه الشاشة للعمل بالمشروع مرة اخرى.

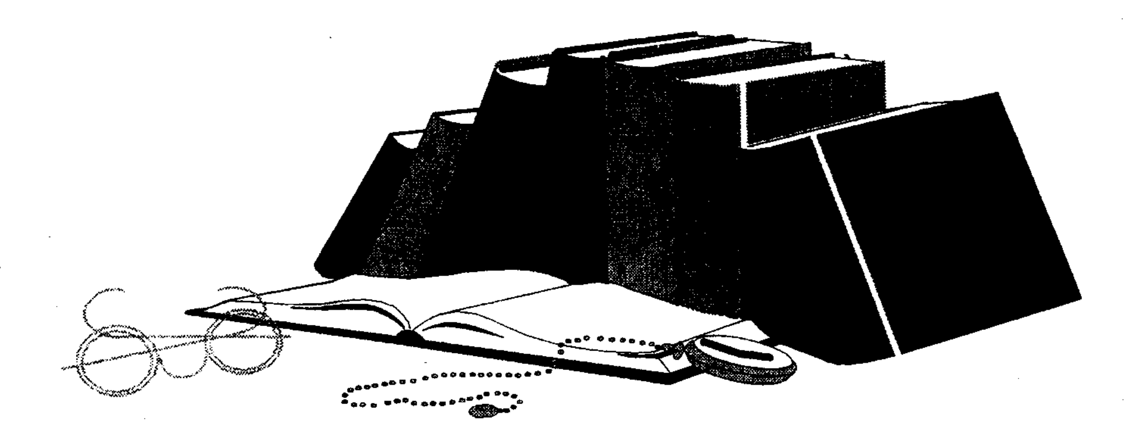
إيعاز Unload

ويتسبب هذا الإيعاز فى مسح هذه الشاشة (الشاشة العاملة Form1) من شاشة الحاسوب وذاكرته ، مع امكانية استدعائها للعمل بالمشروع فى اى وقت يرغب المبرمج فى ذلك ، وبذلك فيعتبر هذا الإيعاز غاية الأهمية فى تضمينه بالروتين الفرعى قبل اظهار الشاشة المستهدفة ، ويمكن كتابة هذا الأمر بالصورة Unload Me ، وبالتالى يأخذ الروتين السابق الصورة التالية:

```
Sub Timer1_Timer ()
    Dim dflvalue
    dflvalue = 30000
      If timer1.Interval = dftvalue Then
        Unload  Me
        Form2.Show 1
      End  If
End Sub
```

ثانيا : طريقة الإنتقال بفعل من المتعلم Click Action

وهنا سوف نحتاج الى وضع زر اوامر عادى Command button او زر اوامر مجسم 3D Command button الى النموذج الذى سوف ننتقل منه الى النموذج التالى بحدث من المتعلم *Click action* مع كتابة الكود الخاص بذلك .



المراجع

Alessi, A.M. & Trolip, S. R. (1985). **Computer-Based Instruction : Methods and Development**. Englewood Cliffs. New Jersey : Prentice-Hall, Inc.

Anderson , C., (1981): **Teaching Computer Literacy, Guidelines for a Six Week Course**, Electronic Learning, Novper.,1981

Anderson , L. W., Scott, C.C., & Hutlock , J. J., (1976) : **The Effects of a Mastery Learning Program on Selected Cognitive, Affective, and Ecological Variables in Grades 1 through 6**. *Paper presented to the annual meeting of the American Educational Research Association,* San Francisco.

Anderson, L. W., and Reynolds, A.(1979) : **The effect of Mastery Learning on the achievement of high ability students and the academic self-concept of low achieving Students**, *Paper presented at the annual meeting of the American Educational Research Association*. San Francisco, California.

Atkinson, R. C., (1968) : **Computer Assisted Instruction in Initial Reading** . *In Proceedings of the 1967 Invitational Conference on Testing Problems*. Princeton; Educational Testing Service.

Allam , Madison ,(1984) **Microcomputer in the Classroom**, Hodder & Stoughton, London.

Al-Hareky, S.M.(1983): **A Study of the effectiveness of Modern Educational Technology on the Mathematics Performance of Elementary Students in Saudi Arabia** ,A Thesis in Curriculum and Instruction, Penn State University.

Alleyway ,B S and Mills ,G. M., (1985) : **Aspects of Educational Technology** , Volume XVIII, Kogan Page, Nichols Publishing Co. New York.

Arthur, Luehrmann , : (1980) : **Computer Illiteracy-A National Crisis and Solution For it** , BYTE, BYTE Publication Inc.June,1980.

Berliner, D. C., Tempus educate, In P. L. Peterson and H. J., Walberg (Eds.), (1979): **Research on Teaching ; Concepts, Findings. and Implications**, Berkeley, Calif., MC-Catchall.

Block, J. H., and Anderson, L. W., (1975) : **Mastery Learning in Classroom Instruction**. New York, Macmillan, Comp.

Bloom, Benjamin S., (1968) : **Learning for Mastery: Evaluation Comment**. *The Center for the Study of Evaluation of Instructional Programs*, Los Angeles, (1)2, May, 1968.

Bloom, Benjamin S., Hastings, Thomas J., and Madaus, George F., (1971) : **Handbook on Formative and Summative Evaluation of Student Learning** . New York, McGraw-Hill Book Comp.

Bloom, B. S., : **Time and Learning**, *American Psychologist*, 1 (29), 1974.

Bloom, Benjamin S., : **Human Characteristics and School Learning**. New York, McGraw-Hill Comp., 1976.

Burns, R. B., (1979) : **Mastery Learning: Does it work?** *Leadership*, 37, 1979.

Burns, P.K. & Bozeman, W.C. (1981). **Computer-Assisted Instruction and Mathematics Achievement** : Is there a relationship? Educational Technology, 21 (10), 32-39.

Baker, F., (1978) : **Computer Managed Instruction, Theory and Practice**, Englewood Cliffs, NJ : Educational Technology Publications Inc.

Backer, H. J., (1982) : **Microcomputers in classroom : dreams and realities**, Baltimore, MD : Center For Social Organization of Schools, Johns Hopkins University, Report No. 218, Jan. 1982.

Beck, J.J. (1979). **The effects on attitudes of anticipated computer assisted instruction in selected high school courses of study,** AEDS, 12 (3), 114-119.

Bitter, Gary G., (1982) : **The Road to Computer Literacy : A Scope and Sequence Model**, Electronic Learning, Sept. 1982.

Bitter, Gary G., (1984) : **Computer in Today's World**, New York : John Wiley and Sons.

Bob Lewis, (1983) : **The Microcomputer and the Teachers Needs,** Computer and Education, Kogan Page, New York.

Bobtail, A., (1982) : **Why Johnny Cant Compute**, Microcomputer, April 1982.

Braun, L., (1981) : **Help! What Computer Should I Buy?** Mathematics Teachers, Nov. 1981.

Blaine, L. & Smith, R.L. (1977). **Intelligent CAI : The role of the curriculum in suggesting Computational models of reasoning**. *Proceedings of the Annual Meetings for Association of Computing Machinery*, Seattle, Washington.

Brown, J.S. & Burton, R.R. (1978). **Diagnostic models for procedural bugs in basic mathematical skills**. Cognitive Science, 2, 155-192.

Burton, R.B. & Brown, J.S. (1979). **An investigation of Computer coaching for informal learning activities**. International Journal of Man-Machine Studies, 11, 5-24.

Chambers, J. A. & Sprecher, J. W., (1983) : **Computer Assisted Instruction : Its use in the classroom**, Prentice-Hall, Englewood Cliffs, NJ.

Christopher, Dean & Quent, Whitlock, (1983) : **A Hand Book of Computer Based Training**, Kogan Page, London.

Coburn, P., P. Kelman, N. Roberts, T. snyder, D. Watt, and C. Weiner, (1982) : **Practical Guide to Computers in Education** , Reading, M A Addison-Wesley.

Cohen, V. B., (1983) : **A Learner-based Evaluation of Microcomputer Software,** Paper Presented at the Annual Meeting of the AERA, Montreal, Canada, April 1983.

Carroll, J. B. (1963) : **A Model of School Learning ,** *Teachers College Record.*

Carroll, J. B., (1973) : **Importance of the Time Factor in Learning** . *Paper Presented at the annual meeting of the American Educational Research Association*, New Orleans.

Chan, K. (1981) : **The Interaction of Aptitude With Mastery Versus Non-mastery Instruction: Effects on Reading Comprehension of Grade three Students** , Unpublished Ph.D. Dissertation University of Western Australia.

Cuban, Larry, Hobos, V . Hansen (1975) : **A study in Organizational response.** *Educational Administration Quarterly*, 1975, (11) 2, 15-37.

Davies, T.P., (1972) : **An Evaluation of Computer-Assisted Instruction Using Drill-and-Practice Programs in Mathematics** , Doctoral Dissertation. United State International University.

Dennis O. Harper & James H. Stewart, (1983) : **RUN : Computer Education,** Brooks/Cole Publishing Company, Monterey, California.

Devin-Sheehan, L., Feldman , R. S., and Allen, V.L., (1976) :**Research on Children Tutoring Children : a Critical Review,** *Review of Educational Research*, 1976, 46, 355-383.

Douglas, J. W. B., (1964) : **The Home and the School: A Study of Ability and Attainment in the Primary School ,** London.

Davies, I.K., (1973) : **Competency Based Learning: Technology Management, and Design,** New York, McGraw-Hill.

Esposito, D., (1973) : **Homogeneous and Heterogeneous Ability Grouping: Principal Findings and Implications For Evaluating and Designing more Effective Educational Environment ,** *Review of Educational Research*, 1973, 163-179.

Ferell, B.G. (1986). **Evaluating the Impact of CAI on Mathematics Learning: Computer immersion project**. Journal of Educational Computing Researching,
2 (3), 76-81.

Frederick, W. C., and Walberg, H. J., (1980) : **Learning as a Function of Time,** *Journal of Educational Research*, 73, 1980.

Finkel, L., (1982) : **Buying a Micro : What Every Educator Should Know**, Electronic Learning, Jan/Feb. 1982.

Fuson, K.C. & Brinko, K.T. (1985). **The comparative effectiveness of microcomputers and flash cards in the drill and practice of basic mathematics facts**. Journal for Research in Mathematics Education, 16 (3), 225-232.

Gangue ,R.M. and L.J. Briggs, (1974) : **Principles of Instructional Design**, New York, Holt, Rinehart and Winston.

Gangue, R.M., W. Wagner, and A. Rojas, (1981) : **Planning and Authoring Computer-Assisted Instruction Lessons**, Educational Technology. September 1981.

Galleon, J.K. (1985): **Instructional conditions for computer based problem solving environment**, *Educational Technology*, 1985, 25, (11), 7-11.

Gary G. Bitter & Ruth A. Camuse, (1984) : **Using a Microcomputer in the classroom**, Reston Publishing Company, Inc., Reston, Virginia.

Gleason, G.T., (1981) : **Microcomputer in Education : The State of the Art**, Educational Technology, Mar. 1981.

Guskey, T. R., and Gates, S. L. (1985) : **A Synthesis of Research on Group-based Master Learning Programs**. *Paper Presented at the annual meeting of the American Educational Research Association*, Chicago, Illinois; 1985.

Hakansson, J., (1981) : **How to Evaluate Educational Courseware**, The Journal of courseware Review, September. 1981.

Heck, W., J. Johnson, and R. Kansky, (1981) : **Guidelines for Evaluating computerized instructional Materials**, Reston, VA : *National Council of Teachers of Mathematics.*

Henderson, R.W. (1985). **Computer-Video instruction in mathematics: Field test of an interactive approach**, *Journal for Research in Mathematics Education*, 16 (3), 207-224.

Hills, John R., **Measurement and Evaluation in the Classroom** , Charles E. Merrill Publishing Comp., A Bell and Howell Comp., Columbus, Ohio, 1976.

Ivor, K. Davies, (1981) : **Instructional Technique** , McGraw-Hill Book Comp., New York.

Jerman, M. (1972). **The use of computers to individualize instruction**, The Mathematics Teacher LXV (5), 395.

Jensen, C. B. (1985). **Using what we have learned in the past to improve the future courseware design**, AEDS, 19 (1), 28-48.

Jerry L. Patterson and Janice H. Patterson, (1983) : **Putting computer Power in Schools : A Step-By-Step Approach**, Prentice-Hall, Inc., New Jersey.

Keller, Fred S., and Sherman, Coilmour J. (1974) : **The Keller Plan Handbook** , Menlo Park, Calif.: W. A., Benjamin.

Kelly, A. V., **Teaching Mixed Ability Classes, An Individualized Approach**, Harper and Row Publishers, London, 1974.

Kolesnik, Walter B., (1970) : **Educational Psychology** , *2nd* Edition, McGraw-Hill Book Comp. New York, 1970, 183-184.

Kelman, P. (1983). **Computers in Teaching Mathematics**, Addison Wesley Publishing Company.

Kearsley, G. P. (1987). **Artificial Intelligence and Instruction: Application and Methods, CAI** : Addison-Wesley Publishing Company.

Kibler , Robert J. & Others (1970) : **Behavioral Objectives and Instruction** , Boston Allen & Bacon ,1970 , 66-76.

Kingman, J., (1981) : **Designing Good Educational Software,** Creative Computing. October, 1981.

Kulik, J.A. & Drowns, R.L. (1984). **Effectiveness of technology in pre-college mathematics science teaching**. Journal of Educational Technology Systems,
12 (2), 137-157.

Kulik, James, A., and Kulik, Chen-Lin C., (1984) : **Effects of Accelerated Instruction on Students** , *Review of Educational Research*, 54(3) , Fall, 1984.

Kulik,, J.A., Bangert, R.L. & Williams G.W. (1983). **Effects of Computer based teaching or secondary school students,** *Journal of Educational Psychology*, 75 (1), 19-26.

Lantz, B.S., Bregar, W.S. & Farely, A.M. (1983). **An Intelligent CAI System for teaching equation solving,** *Journal of Computer-Based Instruction*, 10 (192), 35-42.

Leyton, F. S., (1983) : **The Extent to Which Group Instruction Supplemented by Mastery of the Initial Cognitive Prerequisites Approximates the Learning Effectiveness of One-to-One Tutorial Methods** , Ph.D. Dissertation, University of Chicago, 1983.

Lippitt, Peggy (1975) : **Cross-age Helping Programs** , *Paper Presented at the Annual Meeting of American Educational Research Association*, Washington, D.C. April, 1975.

McGee, Charles S., Kauffman, James M., and Nussen, Judith L. (1977) **Children as Therapeutic Change Agents: Reinforcement Intervention Paradigms,** *Review of Educational Research*, 47, 1977, 451-477.

McNeil, John D., (1977) :**Curriculum: A Comprehensive Introduction** , Little, Brown and Company, Boston, 1977.

Molnar A., (1981) : **The coming of computer Literacy : Are we Prepared for it?**, Educational Technology, Jan. 1981.

Moursand, D., (1981) : **Introduction to Computers in Education for Elementary and Middle School Teachers** , Eugene, OR : International council for computers in Education.

National Council of Teachers of Mathematics, (1984) : **Computers in Mathematics Education**, NCTM, Inc., Yearbook, Virginia, USA.

Norton, P. (1985). **Problem Solving activities in a computer environment: a different angle of vision**. Educational Technology, XXV (11) 36-41.

Nordin, A. B. ,: **Improving Learning : An Experiment in Rural Primary Schools in Malaysia** , *Evaluation Education*, 4, 1980.

Okey, J., and Ciesla, J., : **Mastery Teaching: National Center for the Development of Training Materials in Teacher Education** , Bloomington, Indiana: Indiana University, 1975.

Oshea, T. (1982). **A self-improving quadratic tutor :** *In* D. Sleeman & J.S. Brown (*Ed*), *Intelligent Tutoring Systems*, New York ý: Academic Press

Paprt, Deymout, (1980) : **Mindstorms : Children, Computers, and Powerful Ideas**, New York : Basic Books.

Patton, Robert, and Others, (1981) : **Computer Literacy for All High School Students**, ED 216-679, May 1981

Papert, S. (1980). Mindstorms: **Children, Computers and Powerful Ideas**. New York: Basic Books.

Pagen, J. (1970) INDICOM : **Computer-Assisted Instruction in Operation**, AEDS, 3 (4), 17-28.

Porter, John : **Equality and Education** , *Integrated Education*. (13)76, 1975, 17-20.

Robert P. Taylor (Editor), (1980) : **The Computer in the School : Tutor, Tool, Tutee**. Teachers College, Columbia Un., NY.

Rushby, N., (1984) : **Styles of Computer Based Learning**, In Terry, C. (*Ed*) *Using Microcomputers in Schools*, Crumb Helm, London.

Rosenshine, B. V., :**Content, Time and Direct Instruction**: In PL Peterson and H.J. Warbler, (Eds.), *Research on Teaching: Concepts, Findings, and Implication*. Berkeley, Calf.: McCutchan, 1979.

Stanley, Julian, C. (1976) : **Identifying and Nurturing Mathematically Gifted** , *Phi Delta Kappan*, 58, 1976, 234-237.

Steely, D. (1986) :**Instructional design and CAI**. *In* Harper & Stewart (Ed), *RUN: Computer Education* : Monterey, CA : Brooks/Cole Publishing Company , p. 138-139 .

Suppes, P. (1981) : **University-Level Computer Assisted Instruction at Stanford**: 1968-1980. Stanford, CA : Institute for Mathematical Studies in Social Sciences, Stanford University.

Suppes P., (1974) : **The Place of theory in Educational Research**, *Educational Researcher*, (3) 6, 1974.

Swanson, M. B., and Willis, DJ: **Understanding Exceptional Children and Youth, An Introduction to Special Education** , Rand McNally College Publishing Comp., Chicago, 1979, 110-112.

Watt, D., (1981) : **Computer Literacy : What should schools do about it?**, Instructor, Oct. 1981.

Waybill, H.L., : **Acceleration: Effects, Selection Criteria, and Factors Associated with Achievement** , Scarborough, Ontario, Scarborough Board of *Education Research Department*, 1971.

Webb, Noreen M., and Cullian, Linda K. (1983) : **Group Interaction and Achievement in Small Groups: Stability Overtime**, *American Educational Research Journal*, 20(3), 1983, 411-423.

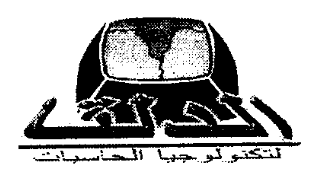
Wenger, E. (1987). **Artificial Intelligence and Tutoring Systems**, Los Altos : Morgan Kaufman Publishers.

Wollenberg, J.P., Handley, H.M. ans Enochs, J.R.. (1985). **Differences in achievement with computer-assisted instruction : Implication for varying student learning styles**, Educational Technology, XXV (11), 51-52.

Wyer, J.A. (1985). New bird on the branch : **Artificial Intelligence and Computer-Assisted Instruction**, PLET, 21 (3) p. 185-191.

بطاقة استفتاء

إن العديد من التحسينات اللازمة لهذه السلسلة تستمد أفكارها من خـــلال رسائل القراء والباحثين الأعزاء ، والتي تلقى لدينا بالغ الاهتمــام ، لذلــك لا تبخل علينا بملاحظاتك ، وتفضل بإرســـالها إلى المؤلــف مباشـــرة علـــي العنوان التالي :



جمهورية مصر العربية . طنطا

الدلتا لتكنولوجيا الحاسبات

٤٠ شارع مسجد الرضوان . طنطا

تليفون : ٣٨١ ٣٣٤٠ (٠٤٠) (٠٠٢) فاكس : ٤٢٩٦ ٣٤٠ (٠٤٠) (٠٠٢)

ص . ب . ٥١٥ طنطا

E-Mail eldelta@future.com.eg

الاســم : .... الوظيفة : ....

العنوان : .... الهاتف : ....

المدينة : .... صندوق بريد : ....

البريد الإلكتروني (E-Mail) : ....

الكتاب الحالي :

[ ] ممتاز [ ] جيد جدا [ ] جيد [ ] مقبول

الإضافات التي تقترحها لهذا الكتاب والموضوعات التي ترغب فيها :

....

....

....

....

أرسلها اليوم من فضلك

كتب للمؤلف

سلسلة
تربويات الحاسوب
استخدام الحاسوب وتكنولوجيا المعلومات في التربية

✓ (١)
تربويات الحاسوب
وتحديات مطلع القرن الحادي والعشرين

✓ (٢)
إعداد وإنتاج
برمجيات الوسائط المتعددة التفاعلية

✓ (٣)
ثقافة الحاسوب
وتكنولوجيا المعلومات

(٤)
تربويات الانترنيت
التعليم بالجلب والتحري والمشاركة

(٥)
الحاسوب
والإبداع الفني والموسيقي

سلسلة
الحاسوب والتحليل الإحصائي للبيانات
باستخدام الحزمة الإحصائية SPSS

✓ (١)
الإحصاء الوصفي
Descriptive Statistics

✓ (٢)
المقارنة بين المتوسطات
Comparison between Means

✓ (٣)
تحليل الانحدار
Regression Analyses

✓ (٤)
التحليل العاملي
Factor Analyses

(٥)
تحليل التمايز والمسار
Discriminate and Pas Analyses